

Penerapan Algoritma Runut-balik pada Penyelesaian Permainan *Countdown Numbers*

Oktavianus Handika - 13515035

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

handikaoktavianus@yahoo.com

Abstraksi—Dalam makalah ini, akan dibahas mengenai cara untuk menyelesaikan permainan teka-teki *Countdown Numbers* menggunakan algoritma runut-balik (*backtracking*). Permainan ini adalah permainan mencari suatu cara berupa ekspresi matematika dari 6 bilangan acak dengan operasi matematika dasar yaitu penjumlahan, pengurangan, perkalian, dan pembagian. Ekspresi matematika yang dibentuk harus sedapat mungkin mencapai atau mendekati bilangan target yang diberikan. Hasil yang dicari lebih difokuskan pada adanya solusi untuk suatu kasus atau tidak.

Kata Kunci—*Countdown Numbers*, *Backtracking*, Teka-teki, Pencarian solusi

I. PENDAHULUAN

Di zaman yang berkembang pesat ini, telah banyak diciptakan permainan-permainan yang beragam. Fungsi permainan saat ini tidak hanya sebagai sarana hiburan tetapi juga sebagai sarana pembelajaran bagi pelajar dan juga masyarakat. Tidak jarang permainan dipertunjukkan pada acara-acara tertentu dari yang bertujuan sebagai acara hiburan bahkan hingga sebagai suatu acara kompetisi.

Salah satu permainan yang ada pada acara televisi adalah "Countdown". Acara televisi ini berasal dari Inggris oleh ITV Studios. Acara ini juga berasal dari acara televisi Prancis "Des Chiffres et Des Lettres". Acara ini merupakan kompetisi antara dua kontestan untuk menyelesaikan setiap teka-teki yang ada.



Gambar 1.1 Title card acara "Countdown"

(Sumber :

[https://en.wikipedia.org/wiki/Countdown_\(game_show\)\)](https://en.wikipedia.org/wiki/Countdown_(game_show)))

Teka-teki tersebut digolongkan menjadi tiga, yaitu "Countdown Letters" di mana kontestan harus membentuk suatu kata dengan karakter sepanjang mungkin dari karakter-

karakter yang ditentukan secara acak, "Countdown numbers" di mana kontestan harus dapat membentuk sebuah ekspresi matematika atau solusi matematis dari bilangan-bilangan yang ditentukan secara acak untuk mencapai suatu bilangan, dan "Countdown Conundrum" di mana kontestan harus dapat menebak sebuah kata yang karakter-karakternya telah diacak.

Pada makalah ini akan dibahas mengenai penyelesaian suatu persoalan pada teka-teki "Countdown Numbers". Penyelesaian persoalan ini menggunakan algoritma runut-balik.

II. DASAR TEORI

A. Permainan *Countdown Numbers*

Permainan *Countdown Numbers* adalah salah satu permainan teka-teki matematika. Dalam permainan ini, terdapat angka-angka yang digolongkan dalam 2 kategori, yaitu "small number" dan "large number". "Small number" adalah kategori angka-angka dari bilangan 1 sampai 10. Angka-angka pada kategori ini dapat muncul paling banyak 2 kali. Sedangkan "large number" adalah kategori angka-angka bilangan yang relatif besar dari "small number", yaitu antara 25, 50, 75, atau 100. Angka-angka pada kategori ini hanya muncul paling banyak 1 kali saja.

Pada permainan ini, pemain dapat memilih berapa banyak angka pada "small number" dan "large number" yang ingin dimainkan dengan banyak angka keseluruhan 6. Setelah menentukan banyak angka tersebut, angka pada kategori "small number" (dan "large number" bila ada) akan dipilih secara acak. Kemudian, suatu bilangan *target* 3-digit akan di-generate secara random.

Tugas pemain dalam permainan ini adalah melakukan operasi bilangan dari 6 bilangan yang ada untuk mencapai bilangan 3-digit tersebut. Operasi hitung yang diperbolehkan adalah penjumlahan, pengurangan, perkalian, dan pembagian. Pemain tidak harus menggunakan seluruh 6 bilangan tersebut untuk mencapai target, namun ke-6 bilangan tersebut hanya dapat dipakai satu kali oleh pemain. Sebagai contoh adalah sebagai berikut.

Pemain memilih 3 “large number” dan 3 “small number”. Bilangan yang terpilih secara acak adalah 100, 75, 25, 9, 4, dan 2. Bilangan *target* yang dibangkitkan adalah 375. Terdapat lebih dari 1 solusi untuk mencapai bilangan *target* tersebut. Contoh solusinya adalah $((9 - 4) * 75)$, $((9 + 4) + 2) * 25$, $((9 - 4) - 2) * 100 + 75$, dsb.

Dalam beberapa saat, terdapat kasus di mana ke-6 bilangan tersebut tidak mungkin mencapai bilangan *target*, contohnya adalah terdapat 6 bilangan terpilih 1, 2, 3, 4, 5, dan 6 serta bilangan *target* 729. Tidak ada satupun solusi yang mencapai bilangan *target*, sehingga diambil satu solusi dengan hasil yang paling mendekati (atau paling kecil selisihnya) dengan bilangan *target*. Dalam contoh ini, solusi yang paling mendekati adalah

$$((6*4*3+1)*5*2) = 730$$

B. Algoritma Runut-balik

Algoritma runut-balik atau yang lebih dikenal dengan Backtracking adalah cara yang metodologis mencoba beberapa sekuens keputusan hingga dicapai suatu solusi untuk suatu permasalahan. Algoritma ini adalah algoritma yang berbasis pada DFS untuk mencari solusi yang lebih mangkus. Saat ini algoritma runut-balik banyak diterapkan pada bidang *artificial intelligence* dan program *games* seperti permainan *tic-tac-toe*, mencari jalan keluar dalam sebuah labirin, catur, dll.

Berikut adalah properti umum yang digunakan metode runut-balik :

1. Solusi persoalan
Solusi dinyatakan sebagai vektor dengan n -tuple
 $X = (x_1, x_2, \dots, x_n)$, $x_i \in$ himpunan berhingga S_i
2. Fungsi pembangkit nilai x_k
Fungsi ini dinyatakan sebagai $T(k)$ dengan $T(k)$ membangkitkan nilai untuk x_k , yang merupakan komponen vektor solusi.
3. Fungsi pembatas (pada beberapa persoalan fungsi ini dinamakan fungsi kriteria). Fungsi ini dinyatakan sebagai $B(x_1, x_2, \dots, x_k)$

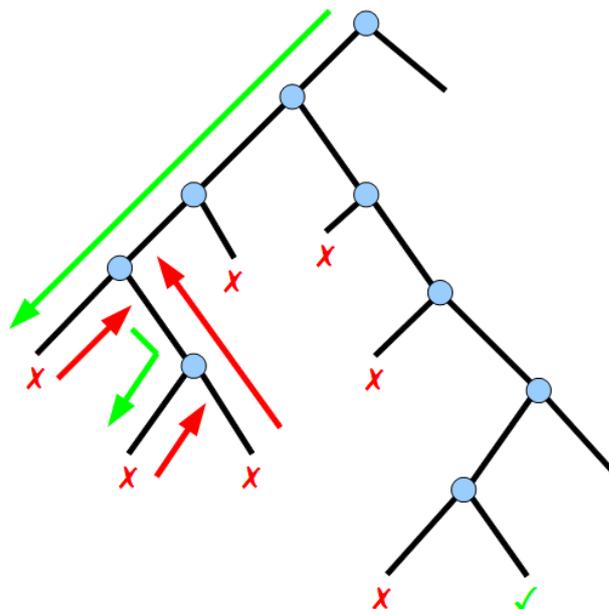
Fungsi pembatas ini menentukan apakah (x_1, x_2, \dots, x_k) mengarah ke solusi. Jika ya, maka pembangkitan nilai untuk x_{k+1} dilanjutkan, tetapi jika tidak, maka (x_1, x_2, \dots, x_k) dibuang dan tidak dipertimbangkan lagi dalam pencarian solusi. Fungsi pembatas tidak selalu dinyatakan sebagai fungsi matematis. Ia dapat dinyatakan sebagai predikat yang bernilai *true* atau *false*, atau dalam bentuk lain yang lebih ekuivalen.

Prinsip pencarian solusi dengan metode runut-balik menggunakan pohon ruang status yang dibangun secara dinamis. Langkah-langkah pencarian solusi adalah sebagai berikut:

1. Solusi dicari dengan membentuk lintasan dan akar ke daun. Aturan pembentukan yang dipakai adalah mengikuti metode pencarian DFS. Simpul-simpul yang

sudah dilahirkan dinamakan **simpul hidup**. Simpul hidup yang sedang diperluas dinamakan **simpul-E** (*expand node*). Simpul dinomoridari atas ke bawah sesuai dengan urutan kelahirannya seperti pada kaidah DFS.

2. Tiap kali simpul-E diperluas, lintasan yang dibangun olehnya bertambah panjang. Jika lintasan yang sedang dibentuk tidak mengarah ke solusi, maka simpul-E tersebut “dibunuh” sehingga menjadi **simpul mati** (*dead node*). Fungsi yang digunakan untuk membunuh simpul-E adalah dengan menerapkan **fungsi pembatas** (*bounding function*). Simpul yang sudah mati tidak akan pernah diperluas lagi.
3. Jika pembentukan lintasan berakhir dengan simpul mati, maka proses pencarian diteruskan dengan membangkitkan simpul anak yang lainnya. Bila tidak ada lagi simpul anak yang dapat dibangkitkan, maka pencarian solusi dilanjutkan dengan melakukan runut-balik ke simpul hidup terdekat (*parent node*). Selanjutnya, simpul ini menjadi simpul-E yang baru. Lintasan baru dibangun kembali sampai lintasan tersebut membentuk solusi.
4. Pencarian dihentikan bila ditemukan sebuah solusi atau tidak ada lagi simpul hidup.



Gambar 2.1 Skema pencarian dengan algoritma runut-balik
(Sumber : <http://www.w3.org/2011/Talks/01-14-steven-phenotype/>)

III. PENERAPAN ALGORITMA PADA PERMAINAN

A. Properti Umum

1. Solusi Persoalan

Misalkan solusi untuk angka dinyatakan dalam vektor X dengan m -tuple dan solusi untuk operatornya dinyatakan dalam vektor Y dengan n -tuple, solusi persoalannya adalah sebagai berikut :

$X = (x_1, x_2, \dots, x_m)$, $x_i \in$ himpunan bilangan yang terpilih
 $Y = (y_1, y_2, \dots, y_n)$, $y_i \in$ himpunan operator sebagai operasi hitung

2. Fungsi pembangkit nilai

Nilai yang dibangkitkan atau di-generate adalah suatu bilangan yang dipilih dan operatornya.

3. Fungsi pembatas

Untuk kasus ini, fungsi pembatasnya adalah fungsi yang menentukan apakah suatu vektor memiliki anggota yang merupakan himpunan bilangan yang terpilih atau tidak :

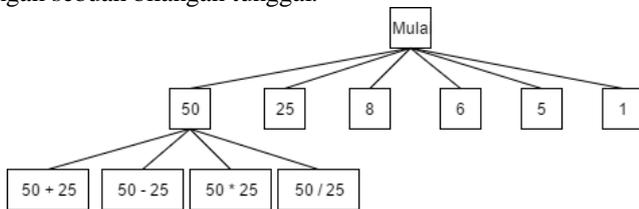
$$X = (x_0, x_1, \dots, x_n) \subseteq \text{himpunan bilangan yang terpilih}$$

Fungsi pembatas ini memaksudkan apabila tidak ada bilangan yang tersisa untuk diambil, maka perluasan simpul suatu state dihentikan.

B. Pohon Ruang Status pada Permainan

Dalam permainan ini, setiap simpul yang terbentuk pada pohon ruang status menggambarkan ekspresi matematika yang terbentuk serta nilai bilangan dari operasi pada ekspresi matematika tersebut. Sebagai contoh, diberikan 6 bilangan acak 50, 25, 8, 6, 5, dan 1, dan bilangan target 445.

Akar pada pohon ruang status akan berupa sebuah ekspresi kosong. Akar tersebut diperluas menjadi sebuah ekspresi dengan sebuah bilangan tunggal.



Gambar 3.1 Contoh Pohon ruang status pada persoalan

Kedalaman pohon ruang status yang terbentuk ditentukan oleh banyaknya bilangan yang dipakai untuk membentuk suatu ekspresi matematika yang memiliki nilai sama dengan bilangan target. Misalkan terdapat 6 bilangan yang tersedia secara acak, kedalaman pohon

Dalam algoritma backtracking, suatu simpul hidup akan diperluas dan pencarian akan dilakukan secara mendalam

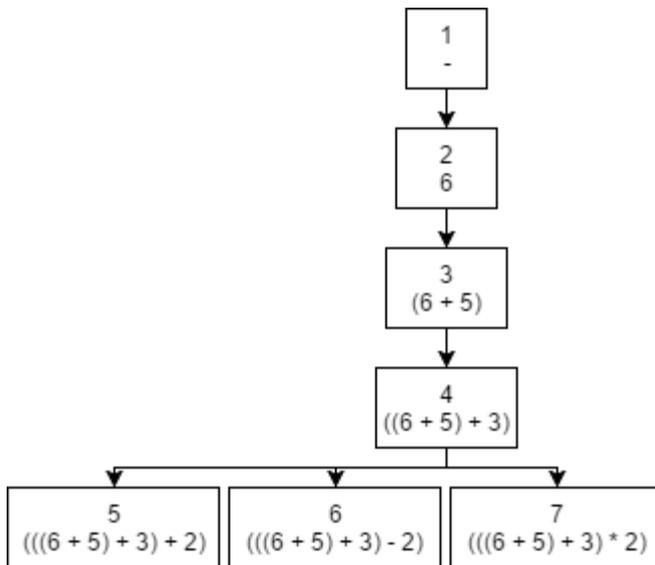
berfokus ada salah satu simpul. Berikut adalah langkah-langkah pembentukan pohon ruang status dari permainan :

1. Ekspresi matematika akan diisi suatu bilangan apabila baru diperluas dari akar pohon ruang status. Bila simpul pada pohon diperluas dari simpul yang sebelumnya telah diperluas, ekspresi matematika pada simpul yang akan diperluas memiliki tambahan berupa suatu operator dan bilangan yang dipilih.
2. Memeriksa apakah simpul mencapai fungsi pembatas atau tidak. Bila belum mencapai fungsi pembatas, simpul diperiksa apakah masih bisa diperluas dengan memasukkan angka yang belum dimasukkan, jika tidak bisa menghidupkan simpul apapun, backtrack ke simpul sebelumnya.
3. Langkah 2 dilakukan terus menerus hingga solusi ditemukan atau tidak ada simpul hidup lagi.

Untuk contoh penggambaran pohon ruang status ini, digunakan contoh sederhana saja, yaitu dengan 4 bilangan acak saja sebagai gantinya dari 6 bilangan. Misalkan terdapat 4 buah bilangan acak 6, 5, 3, dan 2 serta bilangan target 28. Langkah penggambaran pohon ruang status adalah sebagai berikut.

Akar pohon status (simpul 1) berupa ekspresi kosong. Simpul yang diperluas terlebih dahulu adalah state dengan ekspresi matematika dengan bilangan awal 6 sebagai x_1 (simpul 2). Kemudian pencarian solusi diteruskan dengan melakukan operasi hitung ekspresi sebelumnya dengan suatu bilangan. Misalkan prioritas pencarian adalah penjumlahan, pengurangan, perkalian, dan terakhir pembagian. Operasi pembagian hanya dilakukan apabila menghasilkan bilangan tak bersisa ($x_1 \text{ mod } x_2 = 0$).

Pembangkitan simpul berikutnya dilakukan dengan melakukan operasi hitung dengan bilangan yang lainnya. Misalkan diambil bilangan 5 sebagai x_2 dan dilakukan operasi penjumlahan (sehingga $y_1 = '+'$). Simpul baru yang terbentuk (simpul 3) adalah simpul dengan ekspresi matematika $(6 + 5)$, menghasilkan nilai 11 namun belum sama dengan bilangan target. Berikutnya, satu bilangan dari bilangan yang tersisa diambil, misalkan bilangan 3 sebagai x_3 dan dilakukan operasi penjumlahan ($y_2 = '+'$). Simpul baru (simpul 4) terbentuk dengan ekspresi matematika $((6 + 5) + 3)$ menghasilkan nilai 14 dan masih belum sama dengan bilangan target. Pada simpul 5, 6, 7, dan 8, diambil satu bilangan terakhir yang tersisa yaitu 2 sebagai x_4 . Berikutnya, dilakukan operasi penjumlahan hingga pembagian. Operasi-operasi dengan bilangan 2 menghasilkan nilai sebagai berikut beserta simpul yang telah dibangkitkan.



Gambar 3.2 Pohon ruang status yang terbentuk pada contoh persoalan

Pada simpul 5, ekspresi matematika bernilai 16, karena tidak ada lagi bilangan tersisa, maka dilakukan backtrack dari simpul 4 dan diperluas suatu simpul baru (simpul 6) untuk melakukan operasi yang berbeda (yaitu pengurangan) dengan x_4 . Ekspresi matematika pada simpul 6 bernilai 12, karena bernilai tidak sama dengan bilangan *target*, maka dilakukan *backtrack* ke simpul 4 lagi. Pada simpul 7, dilakukan operasi perkalian dengan x_4 dan ekspresi matematikanya bernilai 28 yang mana bernilai sama dengan bilangan *target*. Karena telah didapat ekspresi matematika yang mencapai bilangan *target*, maka pencarian dihentikan.

Jadi, didapat solusi persoalan untuk kasus contoh diatas adalah $X = (6,5,3,2)$ $Y = ('+', '+', '*')$ sehingga menghasilkan ekspresi matematika $((((6 + 5) + 3) * 2) = 28$.

C. Pseudocode Algoritma Runut-balik pada Permainan

Penggunaan algoritma runut-balik yang digunakan adalah algoritma yang menggunakan pengulangan secara rekursif.

Variabel global yang digunakan adalah operator-operator yang akan digunakan selama berjalannya program.

Kamus Global
operator = {'+', '-', '*', '/'}

Pseudocode 1 Variabel global operator

Berikut adalah *pseudocode* keseluruhan prosedur algoritma runut-balik.

```

procedure CariSolusiTarget (input numbers : list of
integer; input target, current_value : integer;
input/output used_numbers : list of integer;
input/output used_operator : list of operator)
{

```

Mencari solusi dari teka-teki "Countdown Numbers" dengan angka-angka yang diberikan dengan algoritma runut-balik

Masukan: numbers dan target terdefinisi, numbers menyatakan angka-angka yang tersedia, target menyatakan bilangan target current_value menyatakan nilai kalkulasi sementara, used_numbers menyatakan list dari angka-angka yang telah dipakai, used_operator menyatakan list dari operator yang digunakan pada setiap langkahnya.

Keluaran: used_operator dan used_numbers yang menyatakan langkah kalkulasi untuk mencapai bilangan target (solusi persoalannya)

Algoritma

```

if (target = current_value)
  CetakSolusi(used_numbers,used_operator)
else
  if (numbers ⊆ ∅)
    {pencarian dihentikan}
  else
    for (number in numbers)
      for (op in operator)
        if ((op = '+') and current_value >=number)
          numbers.remove(number)
          used_numbers.add(number)
          used_operator.add('+')
          CariSolusiTarget (numbers,target,current_
value+number,used_numbers,user_operator)
        else if (op = '-')
          numbers.remove(number)
          used_numbers.add(number)
          used_operator.add('-')

          used_operator.add('-')
          CariSolusiTarget (numbers,target,current_valu
e-number,used_numbers,user_operator)
        else if ((op = '*') and current_value >= number)
          numbers.remove(number)
          used_numbers.add(number)
          used_operator.add('*')
          CariSolusiTarget (numbers,target,current_valu
e*number,used_numbers,user_operator)
        else if ((op = '/') and (number != 0) and
(current_value mod number = 0))
          numbers.remove(number)
          used_numbers.add(number)
          used_operator.add('/')
          CariSolusiTarget (numbers,target,current_val
ue/number,used_numbers,user_operator)
        endif
      endfor
    endif
  endif
endif
endif
endif

```

Pseudocode 2 Prosedur pemecahan teka-teki

Berikut adalah *pseudocode* algoritma untuk mencetak solusi

```

procedure CetakSolusi (input x : list of integer,
input y : list of operator)
{
  Mencetak ekspresi matematika (solusi persoalan)
  Masukan : vektor x dan y yang dinyatakan dalam list
of integer dan list of operator
  Keluaran : solusi persoalan tercetak di layar.
}

```

Kamus Lokal
k : integer

Algoritma
for k ← 1 to length(x) do
 write(x[k])
 if (k != length(x))
 write(y[k])
 endif
endfor

Pseudocode 3 Prosedur untuk mencetak solusi persoalan

IV. HASIL PERCOBAAN IMPLEMENTASI ALGORITMA

Pada pengujian implementasi ini, banyak bilangan dapat bervariasi tidak selalu 6 bilangan acak dan bilangan *target*-nya dapat ditentukan sendiri (tidak harus berupa bilangan 3-digit).

- a. Kasus 4 bilangan acak dan terdapat solusi
Input :

```
28
Masukkan bilangan :
6
5
3
2
```

Output :

```
Bilangan target : 28
Solusi : 6+5+3*2
Waktu : 0 milidetik
```

- b. Kasus 4 bilangan acak namun tidak terdapat solusi
Input :

```
101
Masukkan bilangan :
8
5
5
3
```

Output :

```
Bilangan target : 101
Solusi :
Waktu : 11 milidetik
```

- c. Kasus 5 bilangan acak dan terdapat solusi
Input :

```
144
Masukkan bilangan :
50
9
6
3
2
```

Output :

```
Bilangan target : 144
Solusi : 50*9/6-3*2
Waktu : 1 milidetik
```

- d. Kasus 5 bilangan acak namun tidak terdapat solusi
Input :

```
401
Masukkan bilangan :
25
10
5
3
2
```

Output :

```
Bilangan target : 401
Solusi :
Waktu : 144 milidetik
```

- e. Kasus 6 bilangan acak dan terdapat solusi
Input :

```
720
Masukkan bilangan :
6
5
4
3
2
1
```

Output :

```
Bilangan target : 720
Solusi : 6*5*4*3*2
Waktu : 15 milidetik
```

V. ANALISIS DAN KESIMPULAN

Pada pencarian solusi ini, prioritas yang digunakan (dari yang paling awal hingga paling akhir) adalah penjumlahan, pengurangan, perkalian, dan yang terakhir adalah pembagian. Selain itu, pada percobaan implementasi algoritma, operasi matematika berjalan sekuensial dari kiri ke kanan tanpa memperhatikan prioritas perhitungan pada operasi umumnya. Jika pada suatu persoalan tidak ditemukan solusi, semua simpul pada pohon ruang status akan dikunjungi.

Permainan teka-teki "Countdown Numbers" dapat diselesaikan menggunakan algoritma runut-balik. Bilamana dalam suatu kasus terdapat solusi, algoritma ini akan selalu mendapatkan solusinya. Semakin banyak bilangan yang disediakan, kompleksitas waktunya menjadi berbeda relatif jauh. Algoritma ini tentunya dapat dikembangkan lagi dengan memberi modifikasi untuk improvisasi waktu pencarian solusi.

VI. UCAPAN TERIMA KASIH

Saya mengucapkan terima kasih kepada Tuhan Yang Maha Esa. Karena dengan rahmat-Nya, penulis dapat menyelesaikan makalah "Penerapan Algoritma Runut-balik pada Penyelesaian Permainan Countdown Numbers" dengan baik. Saya juga berterima kasih kepada Dr. Mesayu Leylia Khodra, S.T., M.T., Dr. Nur Ulfa Maulidevi, S.T., M.Sc., dan Dr. Ir. Rinaldi Munir, M.T selaku dosen pengajar mata kuliah IF2211 Strategi Algoritma yang telah membagi ilmunya selama satu semester ini. Saya juga berterima kasih kepada pihak-pihak yang telah membantu selama proses pengerjaan makalah ini.

DAFTAR PUSTAKA

- [1] Munir, Rinaldi. "Diktat Kuliah IF2211 Strategi Algoritma," Program Studi Teknik Informatika Institut Teknologi Bandung, 2009, Bandung, Indonesia
- [2] Numbers, <http://www.huybers.net/backtrack/numbers.html>, diakses pada tanggal 12 Mei 2017, pukul 22.01 WIB
- [3] Letters and Numbers, <http://www.sbs.com.au/ondemand/program/letters-and-numbers>, diakses pada tanggal 12 Mei 2017, pukul 22.30 WIB
- [4] Countdown Numbers Game, <http://resources.woodlands.kent.sch.uk/maths/countdown/index.htm> diakses pada tanggal 14 Mei 2017, pukul 19.43 WIB
- [5] The Computer as Extended Phenotype, <https://www.w3.org/2011/Talks/01-14-steven-phenotype/>, tanggal 15 Mei 2017, pukul 12.05 WIB
- [6] Countdown, [https://en.wikipedia.org/wiki/Countdown_\(game_show\)](https://en.wikipedia.org/wiki/Countdown_(game_show)), diakses pada tanggal 16 Mei 2017, pukul 16.45 WIB
- [7] The Countdown Page, <http://www.thecountdownpage.com/index.htm>, diakses pada tanggal 16 Mei 2017, pukul 17.04 WIB

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, 19 Mei 2017



Oktavianus Handika
13515035