Aplikasi Algoritma *Reverse Greedy* pada *Egyptian Fraction*

Stephen Herlambang

Program Studi Teknik Informatika - Institut Teknologi Bandung
Jl. Ganesha 10 Bandung
e-mail: stephen_herlambang@yahoo.co.id

ABSTRAK

Egyptian fraction **(pecahan Mesir) merupakan cara penulisan pecahan yang digunakan pada sekitar tahun 1650 BC (zaman Mesir Kuno) untuk merepresentasikan pecahan yang memiliki nilai antara 0 dan 1 ($0 < {}^{p}/_{q} < 1$). Inti dari *Egyptian fraction* adalah bagaimana cara merepresentasikan pecahan tersebut sebagai jumlah dari beberapa pecahan unit (pecahan yang memiliki pembilang bernilai 1, disebut juga *unit fraction*) yang berbeda.**
Terdapat berbagai macam metode yang telah dikembangkan untuk memecahkan permasalahan *Egyptian fraction* ini. Salah satu metode yang umum digunakan adalah metode *Greedy*, dimana dalam setiap langkah dicari pecahan unit terbesar yang tidak lebih dari bilangan yang tersisa.
Pada makalah ini akan dibahas kebalikan dari metode *Greedy* yaitu metode *Reverse Greedy*. Metode ini membentuk pecahan Mesir mulai dari nilai terkecil (penyebut terbesar) sampai dengan nilai terbesar (penyebut terkecil). Pembahasan meliputi konsep dan implementasi beberapa varian *Reverse Greedy*, hasil dari beberapa contoh kasus serta perbandingannya, dan analisis terhadap masing-masing metode tersebut.

Kata kunci: *Egyptian fraction*, pecahan Mesir, *Reverse Greedy*, *unit fraction*.

1. PENDAHULUAN

Hieroglyph Mesir kuno memberitahu kita banyak hal tentang orang-orang Mesir kuno, termasuk bagaimana mereka melakukan matematika. *Rhind Mathematical Papyrus*, naskah matematika tertua yang ada, memberitahu kita bahwa sistem bilangan dasar mereka sangat mirip dengan kita kecuali dalam satu hal, yaitu konsep pecahan.

Orang Mesir kuno memiliki cara penulisan angka setidaknya sampai dengan angka 1 juta. Namun, metode mereka dalam menulis pecahan terbatas. Untuk mewakili pecahan ${}^{1}/_{5}$, mereka akan hanya menggunakan simbol untuk 5, dan simbol lain di atasnya. Secara umum, kebalikan dari sebuah integer n ditulis dengan cara yang sama. Mereka tidak punya cara lain untuk menulis pecahan, kecuali untuk simbol khusus seperti untuk ${}^{2}/_{3}$ dan ${}^{3}/_{4}$. Ini tidak berarti bahwa jumlah ${}^{5}/_{6}$ tidak ada di Mesir kuno. Mereka hanya tidak punya cara untuk menulis itu sebagai sebuah simbol. Sebaliknya, mereka akan menulis ${}^{5}/_{6}$ sebagai ${}^{1}/_{2} + {}^{1}/_{3}$.

Jadi, pecahan Mesir adalah sebuah istilah yang sekarang mengacu pada setiap ekspresi dari bilangan rasional sebagai jumlah pecahan unit (*unit fraction*) yang berbeda (pecahan unit adalah kebalikan dari bilangan bulat positif, yaitu pecahan yang pembilangnya bernilai 1).

2. METODE

Metode yang banyak digunakan dan cukup sederhana untuk menyelesaikan permasalahan pecahan Mesir adalah metode *Greedy*, metode ini dalam tiap langkahnya berusaha mencari setiap *unit fraction* yang memiliki jumlah akumulasi tidak lebih dari pecahan yang dicari. Dengan begitu, pecahan Mesir yang terbentuk akan mulai dari nilai terbesar (penyebut terkecil) sampai dengan nilai terkecil (pembilang terbesar).

Kebailkan dari metode ini adalah metode *Reverse Greedy*. Pada metode ini, pecahan Mesir yang terbentuk mulai dari nilai terkecil (penyebut terbesar) sampai dengan nilai terbesar (pembilang terkecil). Pada tiap langkahnya, akan dicari pengembangan (*expansion*) satu pecahan menjadi pecahan unit dan sisanya, dimana sisa tersebut diusahakan memiliki nilai penyebut sekecil mungkin. Langkah ini dilakukan sampai pecahan sisa tersebut adalah pecahan unit juga.

Metode ini memiliki berbagai varian mengingat suatu pecahan biasa dapat memiliki macam representasi pecahan Mesir yang sangat banyak. Makalah ini akan membahas beberapa varian dari metode *Reverse Greedy*, mulai dari *Basic Reverse Greedy*, dan dua varian lainnya.

2.1 Metode *Basic Reverse Greedy*

Metode *Basic Reverse Greedy* dapat dijelaskan sebagai berikut:

Misalkan pecahan yang ingin dicari adalah $^{N}/_{QR}$ dengan Q > 1 dan Q > R. Lalu terdapat dua angka x dan y yang merupakan solusi positif terkecil dari persamaan :

$$Nx - Qy = R \qquad (1)$$

Dengan begitu, akan didapat pecahan $^{N}/_{QR}$ tersebut pengembangan menjadi pecahan unit bernilai $^{1}/_{Qx}$ dan pecahan sisanya bernilai $^{y}/_{Rx}$.

Hal ini dapat ditunjukkan dengan pembuktian sebagai berikut:

$$\frac{N}{QR} = \frac{1}{Qx} + \frac{y}{Rx}$$

$$\frac{Nx}{QRx} = \frac{R + Qy}{QRx}$$

$$Nx = R + Qy \Leftrightarrow Nx - Qy = R$$

Metode *Basic Reverse Greedy* mengambil nilai R = 1, dengan begitu nilai Q = QR, dan pecahan sisa selalu bernilai $^{y}/_{x}$, dimana y selalu bernilai lebih kecil dari x.

Pseudocode dari metode ini adalah sebagai berikut sebagai berikut :

```
procedure BasicReverseGreedy(input N, QR :
integer, output hasil : array of integer)
 Q, R, x, y : integer
 if (N = 0) then {do nothing}
 else if (N = 1) then
  hasil[idx] <- QR
  idx <- idx+1
 else
  Q = QR
  R = 1
  CariXY(N,Q,R,x,y)
  hasil[idx] <- (Q*x)
  idx <- idx+1
  BasicReverseGreedy(y,(R*x),hasil)
```

Algoritma ini merupakan algoritma rekursif dengan hasil representasi pecahan Mesir disimpan pada sebuah *array of integer*. Algoritma ini memiliki basis pengecekan terhadap pembilang yaitu jika pembilang bernilai 0 dan 1.

CariXY merupakan prosedur yang mencari solusi x dan y positif terkecil dari persamaan (1). Konsep dari algoritma ini adalah mencari nilai y terkecil yang menyebabkan nilai x adalah bilangan bulat. Pseudocode untuk prosedur ini adalah sebagai berikut:

```
procedure CariXY(input N, Q, R : integer,
output x, y : integer)
 temp : integer
 k : integer
 found : boolean
 k <- 0
 found <- false;
 while (not found) do
  temp <- R+(Q*k)
  if (temp mod N = 0) then
   x <- temp/N
   y <- k
   found <- true
  else k <- k+1
```

Hasil dari algoritma *Basic Reverse Greedy* ini masih mungkin memiliki dua pecahan unit yang sama. Oleh karena itu setelah algoritma selesai dijalankan masih diperlukan pengecekan terhadap isi *array* yang terbentuk. Pengecekan ini didasarkan pada algoritma *Pairing* pada *Conflict Resolution*. Inti dari algoritma ini adalah bahwa dua pecahan unit yang sama dapat diganti dengan representasi sebagai berikut:

$$\frac{1}{y} + \frac{1}{y} = \begin{cases} \dfrac{2}{y} = \dfrac{1}{y/2}, jika\ y\ genap \\ \dfrac{2}{y+1} + \dfrac{2}{y(y+1)}, jika\ y\ ganjil \end{cases} \qquad (2)$$

Pseudocode untuk pengecekan ini adalah sebagai berikut :

```
procedure CekPairing(input/output hasil :
array of integer)
 cek, i, j : integer
 int cek = hasil[0];
 int i = 0;
 int j;
 while (i < idx) do
  j <- i + 1
  while (j <= idx)
   if (hasil[j] == hasil[i]) then
    if (hasil[j] % 2 == 0) {genap} then
      hasil[i] <- hasil[i]/2
      Delete(hasil,j) {hapus indeks ke-j}
    else
      hasil[i] <- (hasil[i]+1)/2
      hasil[j] <-
(hasil[j]*(hasil[j]+1))/2
    i <- -1  {cek dari awal}
    j <- idx
   else {do nothing}
   j <- j+1
  i <- i+1
```

Hasil dari penggunaan algoritma *Basic Reverse Greedy* terhadap pecahan $^{31}/_{311}$ dapat digambarkan sebagai berikut:

1. Pada tahap pertama, nilai N = 31, dan QR = 311 -> Q = 311 dan R = 1
2. CariXY menghasilkan x = 301 dan y = 30 -> pecahan unit = $^{1}/_{93611}$, sisa = $^{30}/_{301}$
3. Pada tahap kedua, nilai N = 30, dan QR = 301 -> Q = 301 dan R = 1
4. CariXY menghasilkan x = 291 dan y = 29 -> pecahan unit = $^{1}/_{87591}$, sisa = $^{29}/_{291}$
5. Pada tahap ketiga, nilai N = 29, dan QR = 291 -> Q = 291 dan R = 1

6. CariXY menghasilkan x = 281 dan y = 28 -> pecahan unit = $^{1}/_{81771}$, sisa = $^{28}/_{281}$
7. Dan seterusnya sampai N = 2 dan QR = 21 -> Q = 21 dan R = 1
8. CariXY menghasilkan x = 11 dan y = 1 -> pecahan unit = $^{1}/_{231}$, sisa = $^{1}/_{11}$ -> basis

Hasil akhir:

$$\frac{31}{311}=\frac{1}{93611}+\frac{1}{87591}+\frac{1}{81771}+\cdots+\frac{1}{231}+\frac{1}{11}$$

Dari hasil penggunaan metode ini terhadap pecahan $^{31}/_{311}$, dapat dilihat bahwa nilai y secara linear menurun (mulai dari 30 sampai dengan 1, berkurang satu tiap tahap). Karena perubahan nilai y linear, maka nilai jumlah pecahan Mesir yang terbentuk menjadi relatif banyak (pada kasus ini sejumlah 31 buah pecahan unit).

Dengan begitu, secara umum, algoritma *Basic Reverse Greedy* ini kurang efektif. Keuntungan dari *Basic Reverse Greedy* ini adalah tidak akan terbentuknya kalang tidak terbatas pada semua kasus, tanpa diperlukan adanya penyederhanaan pada pecahan sisa.

2.2 Metode *Reverse Greedy* 2

Metode *Reverse Greedy* ini memiliki konsep yang hampir sama dengan *Basic Reverse Greedy*. Perbedaan yang ada adalah dalam pemilihan nilai Q dan R untuk menentukan nilai x, y, dan yang kemudian berpengaruh pada pengembangan pecahan yang bersangkutan.

Pada metode ini, nilai Q dan R diambil dengan mencari nilai Q dan R yang memenuhi nilai (Q-R) terkecil dan nilai Q > 1, serta nilai Q tidak sama dengan nilai R.

Selain itu, pada metode ini, pecahan sisa yang terbentuk harus disederhanakan terlebih dahulu sebelum memasuki perhitungan. Penyederhanaan pecahan ini dapat dilakukan dengan menggunakan pembagian pembilang dan penyebut dengan PBB (Pembagi Bersama ter-Besar) yang dapat diimplementasikan dengan algoritma Euclidian. Penyederhanaan ini dilakukan untuk mencegah terjadinya kalang tidak terbatas pada pencarian nilai x dan y (Hal ini misalnya muncul pada kasus pengembangan pecahan $^{7}/_{15}$).

Pseudocode untuk pencarian nilai Q dan R dari nilai QR adalah sebagai berikut:

```
procedure CariQR(input QR : integer, output
Q, R : integer)
 k : integer
 k <- 1;
 while (k <= sqrt(QR)) do
  if (QR mod k = 0)
   R <- k
   Q <- QR/k
  k <- k+1
```

Pada algoritma ini juga ada kemungkinan memiliki dua pecahan unit yang sama pada hasil yang terbentuk. Oleh karena itu setelah algoritma selesai dijalankan masih diperlukan pengecekan terhadap isi *array* yang terbentuk menggunakan prosedur *CekPairing*.

Psedocode lengkap untuk varian metode *Reverse Greedy* ini adalah sebagai berikut:

```
procedure ReverseGreedy(input N, QR :
integer, output hasil : array of integer)
 Q, R, x, y : integer
 Sederhanakan(N,QR)
 if (N = 0) then {do nothing}
 else if (N = 1) then
  hasil[idx] <- QR
  idx <- idx+1
 else
  CariQR(QR,Q,R)
  CariXY(N,Q,R,x,y)
  hasil[idx] <- (Q*x)
  idx <- idx+1
  ReverseGreedy(y,(R*x),hasil)
```

Hasil dari penggunaan varian algoritma *Reverse Greedy* ini terhadap pecahan $^{31}/_{311}$ dapat digambarkan sebagai berikut:

1. Pada tahap pertama, nilai N = 31, dan QR = 311
2. CariQR menghasilkan nilai Q = 311 dan R = 1
3. CariXY menghasilkan x = 301 dan y = 30 -> pecahan unit = $^{1}/_{93611}$, sisa = $^{30}/_{301}$
4. Pada tahap kedua, nilai N = 30, dan QR = 301
5. CariQR menghasilkan nilai Q = 43 dan R = 7
6. CariXY menghasilkan x = 16 dan y = 11 -> pecahan unit = $^{1}/_{688}$, sisa = $^{11}/_{112}$
7. Pada tahap ketiga, nilai N = 11, dan QR = 112
8. CariQR menghasilkan nilai Q = 14 dan R = 8
9. CariXY menghasilkan x = 2 dan y = 1 -> pecahan unit = $^{1}/_{28}$, sisa = $^{1}/_{16}$ -> basis

Hasil akhir:

$$\frac{31}{311}=\frac{1}{93611}+\frac{1}{688}+\frac{1}{28}+\frac{1}{16}$$

Dari hasil penggunaan metode ini terhadap pecahan $^{31}/_{311}$, dapat dilihat bahwa karena nilai R dicari setinggi mungkin tanpa melebihi nilai Q pada tiap tahap, maka nilai y menurun secara signifikan (mulai dari 30 sampai dengan 1) pada tiap tahapnya. Dengan begitu, jumlah pecahan Mesir yang terbentuk menjadi relatif sedikit (pada kasus ini sejumlah 4 buah pecahan unit).

Hasil dari algoritma ini cukup efektif pada kasus ini, terlihat dari jumlah representasi pecahan Mesir yang relatif sedikit.

2.3 Metode *Reverse Greedy* 3

Metode *Reverse Greedy* ini memiliki konsep yang hampir sama dengan varian *Reverse Greedy* sebelumnya.

Perbedaan yang ada adalah dalam pencarian nilai Q dan R pada prosedur CariQR.

Pada metode sebelumnya, hasil dari pecahan unit yang dihasilkan dapat memiliki lonjakan yang besar (penyebarannya tidak merata). Oleh karena itu, ditambahkan satu syarat lagi dalam pencarian Q dan R yaitu bahwa nilai Q dan R yang dihasilkan memiliki pembagi bersama terbesar sama dengan 1.

Dengan begitu, keseluruhan syarat bagi Q dan R adalah nilai Q dan R yang memenuhi nilai (Q-R) terkecil dan nilai Q > 1, nilai Q tidak sama dengan nilai R, dan PBB(Q, R) = 1.

Pseudocode untuk pencarian nilai Q dan R dari nilai QR adalah sebagai berikut:

```
procedure CariQR(input QR : integer, output
Q, R : integer)
 k : integer
 k <- 1;
 while (k <= sqrt(QR)) do
  if (QR mod k = 0)
   if (FPB(k,(QR/k)) = 0)
    R <- k
    Q <- QR/k
  k <- k+1
```

Penyederhanaan pecahan sebelum perhitungan dengan cara pembagian dengan PBB tetap dilakukan untuk mencegah terjadinya kalang tidak terbatas pada pencarian nilai x dan y. Selain itu, pengecekan terhadap isi *array* yang terbentuk menggunakan prosedur *CekPairing* juga tetap dilakukan karena tetap ada kemungkinan terbentuknya 2 pecahan unit yang sama.

Pseudocode varian metode *Reverse Greedy* ini sama dengan varian sebelumnya.

Hasil dari penggunaan varian algoritma *Reverse Greedy* ini terhadap pecahan $^{31}/_{311}$ dapat digambarkan sebagai berikut:

1. Pada tahap pertama, nilai N = 31, dan QR = 311
2. CariQR menghasilkan nilai Q = 311 dan R = 1
3. CariXY menghasilkan x = 301 dan y = 30 -> pecahan unit = $^{1}/_{93611}$, sisa = $^{30}/_{301}$
4. Pada tahap kedua, nilai N = 30, dan QR = 301
5. CariQR menghasilkan nilai Q = 43 dan R = 7
6. CariXY menghasilkan x = 16 dan y = 11 -> pecahan unit = $^{1}/_{688}$, sisa = $^{11}/_{112}$
7. Pada tahap ketiga, nilai N = 11, dan QR = 112
8. CariQR menghasilkan nilai Q = 16 dan R = 7
9. CariXY menghasilkan x = 5 dan y = 3 -> pecahan unit = $^{1}/_{80}$, sisa = $^{3}/_{35}$
10. Pada tahap keempat, nilai N = 3, dan QR = 35
11. CariQR menghasilkan nilai Q = 7 dan R = 5
12. CariXY menghasilkan x = 4 dan y = 1 -> pecahan unit = $^{1}/_{28}$, sisa = $^{1}/_{20}$ -> basis

Hasil akhir:

$$\frac{31}{311} = \frac{1}{93611} + \frac{1}{688} + \frac{1}{80} + \frac{1}{28} + \frac{1}{20}$$

Dari hasil penggunaan metode ini terhadap pecahan $^{31}/_{311}$, dapat dilihat bahwa karena nilai R dicari setinggi mungkin tanpa melebihi nilai Q pada tiap tahap, maka nilai y juga secara menurun secara signifikan. Tapi karena terdapat tambahan syarat PBB(Q,R) = 1, terdapat perbedaan pada tahap ke-8 (oada saat nilai QR = 112). Pada varian sebelumnya nilai Q dan R adalah 14 dan 8 (PBB(Q,R) > 1), sedangkan pada varian ini nilai Q dan R adalah 16 dan 7 (PBB(Q,R) = 1). Dengan nilai R yang lebih kecil, maka penurunan nilai y juga akan mengecil.

Dengan begitu, jumlah pecahan Mesir yang terbentuk menjadi sedikit lebih banyak dari varian sebelumnya (pada kasus ini sejumlah 5 buah pecahan unit). Tapi, di lain pihak, pecahan unit yang terbentuk memiliki nilai yang penyebaran yang lebih merata. (Jumlah representasi pecahan Mesir lebih sedikit dari varian 1 (*Basic*) dan lebih merata daripada varian 2).

3. HASIL PENGUJIAN

Berikut adalah hasil pengujian beberapa kasus pada implementasi varian-varian algoritma *Reverse Greedy*. Varian 1 adalah *Basic Reverse Greedy*, varian 2 adalah *Reverse Greedy* 2, dan varian 3 adalah *Reverse Greedy* 3.

Tabel 1 Hasil Pengujian untuk Pecahan $^{31}/_{311}$

Varian	Hasil	Jumlah
1	93611, 87591, 81771, 76151, 70731, 65511, 60491, 55671, 51051, 46631, 42411, 38391, 34571, 30951, 27531, 24311, 21291, 18471, 15851, 13431, 11211, 9191, 7371, 5751, 4331, 3111, 2091, 1271, 651, 231, 11	31
2	93611, 688, 28, 16	4
3	93611, 688, 80, 28, 20	5

Tabel 2 Hasil Pengujian untuk Pecahan $^{10}/_{143}$

Varian	Hasil	Jumlah
1	6149, 1247, 435, 15	4
2	99, 65, 63, 35	4
3	99, 65, 63, 35	4

Tabel 3 Hasil Pengujian untuk Pecahan $^{17}/_{180}$

Varian	Hasil	Jumlah
1	9540, 1696, 352, 11	4
2	90, 12	2
3	340, 238, 154, 88, 63, 40, 35	7

Tabel 4 Hasil Pengujian untuk Pecahan $^{23}/_{231}$

Varian	Hasil	Jumlah
1	51051, 46631, 42411, 38391, 34571, 30951, 27531, 24311, 21291, 18471, 15851, 13431, 11211, 9191, 7371, 5751, 4331, 3111, 2091, 1271, 651, 231, 11	23

2	336, 176, 11	3
3	336, 176, 11	3

Tabel 5 Hasil Pengujian untuk Pecahan $^{7}/_{15}$

Varian	Hasil	Jumlah
1	195, 143, 99, 63, 35, 15, 3	7
2	20, 12, 3	3
3	20, 12, 3	3

Tabel 6 Hasil Pengujian untuk Pecahan $^{83}/_{383}$

Varian	Hasil	Jumlah
1	22980, 2220, 518, 70, 5	5
2	22980, 20, 6	3
3	22980, 60, 5	3

Dari hasil pengujian, terlihat bahwa secara umum, jumlah representasi pecahan Mesir yang dihasilkan *Basic Reverse Greedy* lebih banyak dari kedua varian lainnya. Namun, hasil penyebaran pecahan unitnya paling merata.

Hasil pengujian *Reverse Greedy* varian 2 menunjukkan bahwa jumlah representasi pecahan Mesir yang dihasilkan metode ini paling sedikit dibandingkan kedua varian lainnya. Hal ini menunjukkan varian ini paling efektif dari segi jumlah. Namun, pecahan unit yang dihasilkan sering kali memiliki lonjakan yang tidak merata (penyebarannya tidak merata).

Hasil pengujian *Reverse Greedy* varian 3 menunjukkan bahwa jumlah representasi pecahan Mesir yang dihasilkan metode ini secara umum lebih sedikit dari varian 1 dan sama atau sedikit lebih banyak dari varian 2. Dari segi penyebaran pecahan unit yang dihasilkan, varian ini relatif lebih baik dari varian 2, meskipun tidak semerata varian 1.

4. KESIMPULAN

Secara umum, algoritma *Basic Reverse Greedy* kurang efektif karena jumlah representasi *Egyptian Fraction* yang dihasilkan relatif lebih banyak. Keuntungan dari *Basic Reverse Greedy* ini adalah tidak diperlukan adanya penyederhanaan pada pecahan sisa untuk mencegah terbentuknya kalang tidak terbatas. Selain itu, penyebaran hasil pecahan unit juga relatif lebih merata.

Secara umum, algoritma *Reverse Greedy* varian 2 cukup efektif karena jumlah representasi pecahan Mesir yang dihasilkan relatif sedikit. Namun, hasil dari pecahan unit yang dihasilkan dapat memiliki penyebaran yang tidak merata.

Secara umum, algoritma *Reverse Greedy* varian 3 cukup efektif. Hal ini karena jumlah representasi pecahan Mesir yang dihasikan relatif sedikit dan penyebarannya juga cukup merata. Jumlah representasi pecahan Mesir lebih sedikit dari *Basic Reverse Greedy* dan lebih merata daripada *Reverse Greedy* varian 2.

Terdapat berbagai varian metode *Reverse Greedy* yang dapat diimplementasikan. Hal yang patut diperhatikan adalah bagaimana menghasilkan nilai Q dan R (dekomposisi penyebut) yang sesuai untuk dapat mencapai hasil yang diharapkan.

REFERENSI

[1] Eppstein, David. "Egyptian Fractions". http://www.ics.uci.edu/~eppstein/numth/egypt/ (Waktu akses : 1 Januari 2010, pukul 14.13)

[2] Gong, Kevin. "Egyptian Fractions". http://kevingong.com/Math/EgyptianFractions.pdf (Waktu akses : 1 Januari 2010, pukul 20.23)

[3] *MathPages*. http://www.mathpages.com/home/inumber.htm (Waktu akses : 1 Januari 2010, pukul 13.32)

[4] Weisstein, Eric W. "Egyptian Fraction". http://mathworld.wolfram.com/EgyptianFraction.html (Waktu akses : 1 Januari 2010, pukul 13.23)