

MENINGKATKAN KECEPATAN KOMPUTASI UNTUK PENGAMBILAN KEPUTUSAN (KLASIFIKASI) MELALUI REDUKSI DIGIT NUMERIK TAK SIGNIFIKAN

Kuspriyanto, Samiran, Tri Aulat Junarwoto
Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung
kuspriyanto@yahoo.com samiran_smg@yahoo.com triajunarwoto@yahoo.com

Abstrak

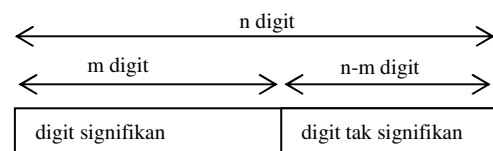
Kecepatan atau durasi komputasi akan sangat berpengaruh pada kecepatan proses pengambilan keputusan. Dua pendekatan yang sering dilakukan untuk mempercepat proses komputasi adalah dengan memparalelkan prosesor (multiprosesor) dan dengan membatasi (mereduksi) proses komputasi. Meskipun multiprosesor dapat mempercepat penyelesaian proses komputasi secara keseluruhan, namun kecepatan masing-masing proses tetap tidak berubah. Kecepatan masing-masing proses tergantung pada algoritma yang digunakan. Pembatasan proses komputasi merupakan algoritma alternatif yang dapat digunakan untuk mempercepat penyelesaian proses komputasi. Dengan algoritma ini, sebuah komputasi yang seharusnya memproses bilangan n digit untuk ketelitian maksimum, dapat direduksi dengan hanya memproses m digit, dengan $m < n$. Dengan hanya memproses m digit dari n digit yang ada, penyelesaian komputasi menjadi lebih cepat tetapi tingkat ketelitian akan berkurang. Sampai batas tertentu, berkurangnya tingkat ketelitian tidak akan menjadi masalah, yaitu apabila keputusan yang diambil berdasarkan hasil komputasi tereduksi tersebut masih tetap sama. Hal ini dimungkinkan jika digit-digit yang tidak diproses (dihilangkan) adalah digit-digit yang tak signifikan. Paper ini akan membahas digit-digit mana yang disebut sebagai digit tak signifikan sehingga dapat diabaikan dalam proses komputasi numerik.

Kata kunci: reduksi, digit numerik, digit tak signifikan, klasifikasi.

I. Pendahuluan

Pemrosesan data komputer (*computer data processing*) umumnya terdiri atas proses-proses komputasi data numerik, data non numerik, serta proses pengambilan keputusan. Ada dua hal utama untuk meningkatkan proses komputasi yaitu dengan memparalelkan pemrosesan (menggunakan komputer multiprosesor), dan dengan mereduksi jumlah digit numerik setiap operand. Komputer multiprosesor memang dapat mempercepat penyelesaian proses komputasi karena banyak tugas akan dikerjakan dalam waktu yang bersamaan[1], tetapi ini tidak berarti mempercepat masing-masing proses. Sebaliknya, dengan reduksi digit numerik tak signifikan pada *operand* akan dapat meningkatkan kecepatan masing-masing proses. Mereduksi digit-digit tak signifikan berarti akan memperpendek waktu penyelesaian komputasi namun mengurangi tingkat ketelitian hasilnya. Kebanyakan para perancang pengambil keputusan kurang memiliki dasar yang kuat dalam menentukan berapa digit ketelitian yang diperlukan dalam perhitungan numerik. Umumnya hanya berpedoman bahwa semakin banyak digit yang diproses akan semakin baik hasil yang

diperoleh, dan semakin *valid* (benar) keputusan yang diambil. Penelitian di sini akan menunjukkan bahwa suatu komputasi yang harus memproses n digit untuk memperoleh ketelitian maksimum, sebenarnya dalam banyak hal, cukup hanya memproses m digit ($m < n$), untuk memperoleh hasil komputasi yang *valid* dan *acceptable*. Ini berarti sejumlah $n-m$ digit tidak perlu diproses, dan hal ini tidak menjadi masalah apabila $n-m$ digit tersebut merupakan digit-digit yang tidak signifikan (Gambar 1).

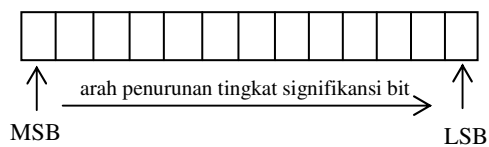


Gambar 1. Posisi digit tak signifikan dalam sebuah bilangan

II. Digit Tak Signifikan

Setiap bilangan memuat tingkatan signifikansi digit-digitnya. Digit paling kiri (paling depan) merupakan digit dengan nilai signifikansi tertinggi, yang sering disebut dengan *Most Significant Bit* (MSB). Digit-digit berikutnya,

semakin ke kanan semakin menurun tingkat signifikansinya, dan digit terakhir merupakan digit dengan nilai signifikansi terendah yang disebut *Least Significant Bit* (LSB) (Gambar 2). Sebuah komputasi yang memproses sejumlah *operand* dengan n bit (digit), akan menghasilkan hasil komputasi dengan ketelitian maksimum jika seluruh digit diproses, dan waktu yang digunakan juga maksimum. Mengabaikan digit-digit yang tidak signifikan akan mengurangi beban komputasi.



Gambar 2. Tingkat signifikansi digit pada suatu bilangan

Signifikansi suatu digit dalam suatu operand tergantung pada validitas hasil komputasi. Validitas hasil komputasi bergantung pada validitas keputusan yang diambil berdasarkan hasil komputasi tersebut. Apabila suatu digit dalam suatu operand diproses maupun tidak diproses ternyata tidak mempengaruhi keputusan yang diambil maka digit tersebut dikatakan tak signifikan. Digit-digit inilah yang dapat direduksi untuk mempercepat penyelesaian dan mengurangi beban komputasi.

III. Menentukan Digit Tak Signifikan

Untuk dapat mengetahui digit-digit tak signifikan suatu operand, perlu mengamati proses perhitungan hingga mendapatkan hasil. Hasil dari perhitungan ini diklasifikasikan dalam kelas-kelas interval. Jika suatu hasil proses perhitungan pada *operand* dengan pendekatan (pembulatan) n digit sama hasil klasifikasinya dengan yang menggunakan pendekatan m digit ($m < n$), maka $n - m$ digit ini dikatakan digit tidak signifikan. Sebagai contoh suatu perhitungan harus memproses hingga 10 digit untuk mendapat klasifikasi A. Jika hanya dengan memproses 8 digit yang pertama saja telah dapat dikategorikan dalam klasifikasi A, maka 2 digit terakhir dikatakan sebagai digit-digit tak signifikan. Penulis mengamati hasil-hasil percobaan perhitungan numerik pada bilangan random. Ada 3 jenis percobaan yang dilakukan, yaitu:

1. Menghitung nilai $\sum_{i=1}^{30} a_i$ dengan a bilangan random terbatas 0 sampai 10.
2. Menghitung nilai $\sum_{i=1}^{30} a_i b_i$ dengan a dan b bilangan random terbatas 1 sampai 5.
3. Menghitung keliling lingkaran ($K = \pi d$) dengan pembulatan nilai π yang bervariasi dan d bilangan random.

Secara rinci ketiga percobaan di atas dapat dijelaskan sebagai berikut.

1. Menghitung nilai $\sum_{i=1}^{30} a_i$ dengan a bilangan random terbatas 0 sampai 10

Prosedurnya dilakukan seperti berikut.

- a. Menentukan 30 bilangan random terbatas 0 sampai dengan 10 menggunakan program aplikasi *Microsoft Excel* 2003.
- b. Mengelompokkan ke 30 bilangan random tersebut ke dalam 16 kelompok pembulatan, yaitu pembulatan 0 hingga 15 digit.
- c. Menjumlahkan ke 30 bilangan yang telah dikelompokkan menurut pembulatan tersebut pada b, yang secara matematis dinotasikan dengan $\sum_{i=1}^{30} a_i$ dengan a adalah bilangan random terbatas 0 sampai 10.
- d. Penjumlahan dilakukan sebanyak 100 kali, tiap set penjumlahan terdiri dari 30 bilangan random, hingga didapatkan 100 data hasil penjumlahan.
- e. Seratus data hasil penjumlahan yang didapatkan, dikelompokkan ke dalam 6 kelas, 8 kelas, 10 kelas, 12 kelas, 15 kelas, 20 kelas, dan 22 kelas.
- f. Hasil-hasil pengelompokkan ke dalam kelas-kelas tersebut, ditampilkan dalam bentuk tabel dan grafik distribusi frekwensi, baik yang terdistribusi dalam 6 kelas, 8 kelas, 10 kelas, 12 kelas, 15 kelas, 20 kelas, maupun 22 kelas. Dengan catatan semakin banyak kelasnya, semakin kecil rentang interval tiap kelasnya. Salah satu tabel dan grafik dapat dilihat pada Tabel 2 dan Gambar 2 yang mengelompokkan data ke dalam 12 kelas dengan rincian kelas seperti terlihat pada Tabel 3.

- g. Dari 100 set bilangan random yang dijumlahkan, salah satunya seperti terlihat pada Tabel 1.

Tabel 1. Daftar 30 bilangan random dengan pembulatan 0 hingga 15 digit

Data ke	Banyaknya digit pembulatan					
	0	1	2	3	...	15
1	9	9.2	9.19	9.190	...	9.189568169270070
2	1	1.3	1.33	1.327	...	1.326750085475180
3	2	2.1	2.13	2.126	...	2.125859451776870
4	0	0.2	0.18	0.179	...	0.178945900686844
5	10	9.5	9.54	9.541	...	9.540960645498360
6	4	3.9	3.89	3.888	...	3.887769439434820
7	4	3.9	3.90	3.897	...	3.896934689329130
8	4	3.9	3.91	3.909	...	3.908853901343290
9	1	1.0	0.97	0.967	...	0.967435092428459
10	3	3.3	3.29	3.293	...	3.293312739897350
11	6	5.7	5.74	5.738	...	5.737688531841260
12	2	2.0	1.95	1.952	...	1.952341212999610
13	6	6.1	6.06	6.056	...	6.056322869157260
14	1	1.3	1.26	1.260	...	1.260167993408290
15	8	8.2	8.22	8.223	...	8.223133262814990
16	2	1.5	1.51	1.505	...	1.505044106038780
17	3	3.2	3.15	3.151	...	3.151435406834660
18	2	2.4	2.36	2.360	...	2.360400523212780
19	4	3.6	3.61	3.607	...	3.606808472756430
20	5	4.6	4.58	4.579	...	4.579280940864310
21	5	5.2	5.23	5.231	...	5.231390228386670
22	10	9.6	9.60	9.597	...	9.596702985440320
23	1	1.3	1.33	1.333	...	1.333290009400920
24	6	6.4	6.35	6.354	...	6.354452326460570
25	4	3.7	3.69	3.692	...	3.6917437826885140
26	9	8.7	8.73	8.729	...	8.728959750712250
27	9	9.2	9.23	9.227	...	9.226981745057930
28	0	0.3	0.30	0.301	...	0.300988213968756
29	8	8.1	8.12	8.119	...	8.118519387151280
30	4	4.1	4.09	4.092	...	4.092171212716170
Jml	133	133.5	133.44	133.423	...	133.424213077049000

Kolom ke-2 pada Tabel 1 adalah 30 bilangan random dengan pembulatan 0 digit, kolom ke-3, ke-4, ke-5 dan seterusnya hingga kolom ke-16 masing-masing adalah 30 bilangan random yang sama dengan bilangan pada kolom ke-2, dengan pembulatan 1 digit, 2 digit, 3 digit, dan seterusnya hingga pembulatan 15 digit. Sedangkan baris terakhir pada tabel tersebut menunjukkan jumlah tiap set bilangan random yang telah dikelompokkan ke dalam pembulatan yang sama.

Pada Tabel 2, kolom pertama menyatakan jenis pembulatan bilangan random, yakni 0 hingga 15 digit. Sedangkan kolom kedua hingga kolom ke 13 menyatakan kelas ke-1, kelas ke-2, kelas ke-3 dan seterusnya sampai kelas ke-12. Interval tiap kelas ditentukan seperti pada Tabel 3. Angka-angka distribusi yang terdapat pada kolom ke-2 hingga kolom ke-13 pada Tabel 2 menyatakan frekuensi kemunculan jumlah 30 bilangan random dalam kelas-kelas yang bersesuaian

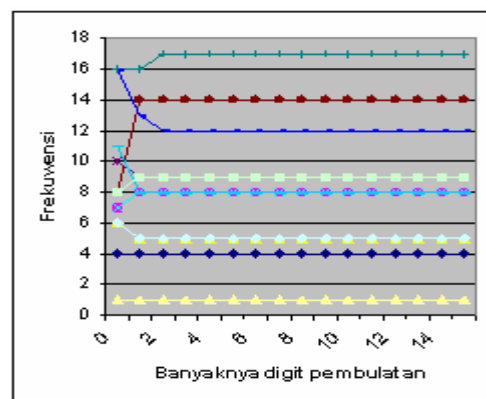
Tabel 2. Frekuensi kemunculan jumlah 30 bilangan random pada tiap interval kelas dalam 12 kelas (Percobaan 1)

Digit pemb	Kelas ke											
	1	2	3	4	5	6	7	8	9	10	11	12
0	4	7	6	7	10	8	16	16	11	6	8	1
1	4	8	5	8	9	14	16	13	8	5	9	1
2	4	8	5	8	9	14	17	12	8	5	9	1
3	4	8	5	8	9	14	17	12	8	5	9	1
4	4	8	5	8	9	14	17	12	8	5	9	1
5	4	8	5	8	9	14	17	12	8	5	9	1
6	4	8	5	8	9	14	17	12	8	5	9	1
7	4	8	5	8	9	14	17	12	8	5	9	1
8	4	8	5	8	9	14	17	12	8	5	9	1
9	4	8	5	8	9	14	17	12	8	5	9	1
10	4	8	5	8	9	14	17	12	8	5	9	1
11	4	8	5	8	9	14	17	12	8	5	9	1
12	4	8	5	8	9	14	17	12	8	5	9	1
13	4	8	5	8	9	14	17	12	8	5	9	1
14	4	8	5	8	9	14	17	12	8	5	9	1
15	4	8	5	8	9	14	17	12	8	5	9	1

Tabel 3. Daftar interval dalam 12 kelas nilai $\sum_{i=1}^{30} a_i$

Kelas ke	Interval kelas	Kelas ke	Interval kelas
1	$0 \leq X < 125$	7	$150 \leq X < 155$
2	$125 \leq X < 130$	8	$155 \leq X < 160$
3	$130 \leq X < 135$	9	$160 \leq X < 165$
4	$135 \leq X < 140$	10	$165 \leq X < 170$
5	$140 \leq X < 145$	11	$170 \leq X < 190$
6	$145 \leq X < 150$	12	$190 \leq X < 300$

Secara grafis, Tabel 2 dapat digambarkan sebagaimana Gambar 3. Setiap garis menggambarkan angka-angka dari setiap kolom.



Gambar 3. Grafik frekuensi kemunculan jumlah 30 bilangan random pada tiap interval kelas dalam 12 kelas (Percobaan 1)

Tabel dan grafik di atas menunjukkan bahwa pada pembulatan 0 digit, 1 digit, dan 2 digit, angka distribusi masih berubah. Namun mulai pembulatan 2 digit hingga 15 digit, angka distribusi selalu konstan (tetap) pada setiap interval kelas. Meskipun tidak ditunjukkan di sini, hal ini terjadi pula pada pengelompokan kelas yang terdiri dari 8 kelas, 10 kelas, 15 kelas, 20 kelas maupun 22 kelas. Ini menandakan bahwa untuk operasi penjumlahan dengan operand bilangan random, kita hanya memerlukan pembulatan hasil sampai 2 digit saja. Digit ke-3, ke-4, ke-5, dan seterusnya hingga digit ke-15 tidak diproses pun tidak akan mengubah hasil klasifikasi mengingat angka distribusi selalu sama. Dapat disimpulkan bahwa digit ke-3 hingga digit ke-15 merupakan digit tak signifikan.

2. Menghitung nilai $\sum_{i=1}^{30} a_i.b_i$ dengan a dan b bilangan random terbatas 1 sampai 5

Prosedur yang dilakukan seperti berikut.

a. Menentukan 30 pasang bilangan random terbatas 1 sampai dengan 5, menggunakan program aplikasi *Microsoft Excel* 2003.

b. Menghitung nilai dari $\sum_{i=1}^{30} a_i.b_i$ dengan a

dan b adalah dua bilangan random berbeda sebagaimana ditentukan dalam poin a, yang masing-masing dengan pembulatan 0 dan 0 digit, 0 dan 1 digit, 1 dan 1 digit, 1 dan 2 digit, 2 dan 2 digit, 2 dan 3 digit, 3 dan 3 digit, 3 dan 4 digit, 4 dan 4 digit, 4 dan 5 digit, serta 5 dan 5 digit. Perlu diperhatikan bahwa jika kita mengalikan dua bilangan yang masing-masing dengan pembulatan m dan n digit maka hasil dari perkalian tersebut menggunakan pembulatan $m+n$ digit. Sedangkan untuk penjumlahan sejumlah operand dengan pembulatan n digit akan menghasilkan hasil dengan pembulatan n digit pula.

c. Pengoperasian seperti pada poin b dilakukan sebanyak 35 kali, pada set bilangan random yang berbeda.

d. Hasil dari 35 kali pengoperasian pada poin c dikelompokkan ke dalam 8 kelas, 10 kelas, 12 kelas, 16 kelas, dan 20 kelas interval, dengan pengertian, semakin banyak jumlah kelas berarti semakin kecil rentang setiap kelasnya.

e. Hasil pengelompokan seperti pada poin d ditampilkan dalam bentuk tabel dan grafik, baik yang terdistribusi dalam 8 kelas, 10 kelas, 12 kelas, 16 kelas, maupun 20 kelas. Sebagai contoh, salah satu tabel dan grafik dapat dilihat pada Tabel 5 dan Gambar 4.

Tabel 5 dan grafik pada Gambar 4 merepresentasikan pengelompokan data

distribusi nilai $\sum_{i=1}^{30} a_i.b_i$ ke dalam 10 kelas, dengan pembagian kelas seperti terlihat pada Tabel 4.

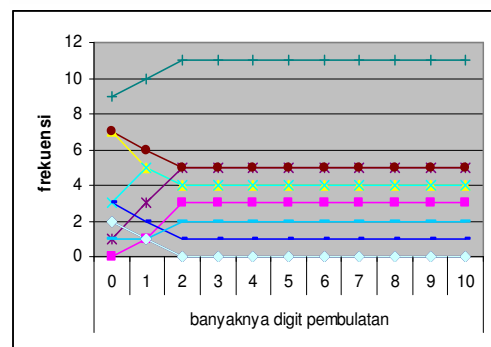
Tabel 4. Daftar interval dalam 10 kelas nilai

$$\sum_{i=1}^{30} a_i.b_i$$

Kelas ke	Interval kelas	Kelas ke	Interval kelas
1	227 ≤ X < 237	7	277 ≤ X < 287
2	237 ≤ X < 247	8	287 ≤ X < 297
3	247 ≤ X < 257	9	297 ≤ X < 307
4	257 ≤ X < 267	10	307 ≤ X < 317
5	267 ≤ X < 277	11	317 ≤ X < 328

Tabel 5. Frekuensi kemunculan jumlah 30 bilangan random pada tiap interval kelas dalam 10 kelas (Percobaan 2)

digit pemb.	kelas ke									
	1	2	3	4	5	6	7	8	9	10
1	2	1	0	0	0	0	0	0	0	0
2	0	1	3	3	3	3	3	3	3	3
3	7	5	4	4	4	4	4	4	4	4
4	3	5	4	4	4	4	4	4	4	4
5	1	3	5	5	5	5	5	5	5	5
6	7	6	5	5	5	5	5	5	5	5
7	9	10	11	11	11	11	11	11	11	11
8	3	2	1	1	1	1	1	1	1	1
9	1	1	2	2	2	2	2	2	2	2
10	2	1	0	0	0	0	0	0	0	0



Gambar 4. Grafik frekuensi kemunculan jumlah 30 bilangan random pada tiap interval kelas dalam 10 kelas (Percobaan 2)

Melalui Tabel 5 dan grafik pada Gambar 4, terdapat kenyataan bahwa perubahan distribusi hanya terdapat pada pembulatan dengan 0 hingga 2 digit. Mulai pembulatan dengan 2 digit hingga 16 digit, angka distribusi selalu sama. Hal ini terjadi pula pada pengelompokan dalam 8 kelas, 12 kelas, 16 kelas maupun 20 kelas, yang mengingat keterbatasan tempat, tidak seluruh tabel dan grafik disertakan dalam paper ini. Dari hasil-hasil ini dapat disimpulkan bahwa dalam kasus gabungan operasi perkalian dan penjumlahan hanya memerlukan pembulatan hasil hingga 2 digit saja.

3. Menghitung keliling lingkaran ($K = \pi d$) dengan pembulatan nilai π yang bervariasi dan d bilangan random

Prosedur yang dilakukan adalah sebagai berikut.

- Menentukan 30 bilangan random terbatas 0 hingga 20 sebagai panjang diameter lingkaran.
- Menghitung keliling masing-masing lingkaran dengan nilai π yang bervariasi jumlah digit pembulatannya, mulai dengan 0 digit, 1 digit, 2 digit dan seterusnya hingga 10 digit di belakang koma.
- Mengelompokkan hasil penghitungan keliling lingkaran tersebut ke dalam 4 kelas, 8 kelas, 16 kelas, dan 22 kelas.
- Hasil pengelompokan disajikan dalam format tabel dan grafik, baik yang terdistribusi dalam 4 kelas, 8 kelas, 16 kelas, maupun 22 kelas. Salah satu tabel dan grafik adalah Tabel 7 dan grafik pada Gambar 5.

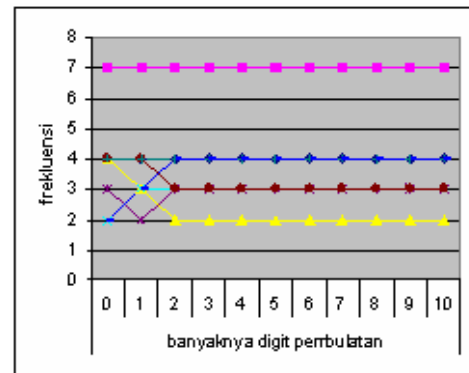
Tabel 7 dan Gambar 5 merepresentasikan distribusi data keliling lingkaran dengan nilai π yang bervariasi pembulatannya, mulai dari pembulatan 0 hingga 10 digit, yang diklasifikasikan ke dalam 8 kelas interval. Pembagian interval dalam 8 kelas tersebut dapat dilihat pada Tabel 6.

Tabel 6. Daftar interval dalam 10 kelas data keliling lingkaran

Kelas ke	Interval kelas	Kelas ke	Interval kelas
1	$0 \leq X < 8$	7	$32 \leq X < 40$
2	$8 \leq X < 16$	8	$40 \leq X < 48$
3	$16 \leq X < 24$	9	$48 \leq X < 56$
4	$24 \leq X < 32$	10	$56 \leq X < 64$

Tabel 7. Frekuensi kemunculan 30 keliling lingkaran pada tiap kelas dalam 8 kelas (Percobaan 3)

digit pemb.	kelas ke							
	1	2	3	4	5	6	7	8
0	4	7	4	2	3	4	4	2
1	4	7	3	3	2	4	4	3
2	4	7	2	3	3	3	4	4
3	4	7	2	3	3	3	4	4
4	4	7	2	3	3	3	4	4
5	4	7	2	3	3	3	4	4
6	4	7	2	3	3	3	4	4
7	4	7	2	3	3	3	4	4
8	4	7	2	3	3	3	4	4
9	4	7	2	3	3	3	4	4
10	4	7	2	3	3	3	4	4



Gambar 5. Grafik Frekuensi kemunculan 30 keliling lingkaran pada tiap interval kelas dalam 8 kelas (Percobaan 3)

Melalui Tabel 7 dan grafik pada Gambar 5 dapat dilihat bahwa ternyata hasil perhitungan keliling lingkaran mulai konstan pada pembulatan nilai π dengan 1 digit. Namun pada klasifikasi yang terdiri dari 16 kelas dan 22 kelas (tidak disertakan dalam paper ini), angka distribusi konstan mulai pembulatan nilai π dengan 2 digit. Ini menunjukkan bahwa pada perhitungan fungsi seperti kasus menghitung keliling lingkaran yang diameternya bilangan random, hanya diperlukan pembulatan nilai π hingga 2 digit, mengingat tidak ada perbedaan data distribusi antara pembulatan nilai π dengan 2 digit, 3 digit, 4 digit dan seterusnya.

IV. Analisis dan Kesimpulan

Pada dasarnya komputasi numerik yang banyak dibutuhkan adalah berbentuk penjumlahan dan/atau perkalian, dengan operan bersifat random terbatas. Hasil perhitungan ini yang umumnya dipergunakan untuk menentukan pilihan melalui klasifikasi

(suatu bentuk pengambilan keputusan), ternyata operan-operannya cukup hanya dengan pembulatan 2 digit. Ketelitian dengan pembulatan 3 digit atau lebih tidak akan bermakna mengingat hasil yang didapat tidak mempengaruhi keputusan yang diambil. Ini berarti bahwa digit-digit ketiga, keempat, kelima dan seterusnya dapat dikatakan sebagai digit-digit yang tidak signifikan. Dengan mereduksi (mengabaikan) digit-digit yang tidak signifikan ini, maka jumlah operasi menjadi lebih sedikit dan proses komputasi menjadi lebih cepat dibanding dengan komputasi yang memproses keseluruhan digit yang ada. Percepatan komputasi yang diperoleh akan sebanding dengan reduksi digit yang dapat dilakukan. Selanjutnya, kenyataan ini akan membuka peluang bagi para perancang arsitektur prosesor dan bahasa pemrograman untuk membangun sistem komputer yang dapat memfasilitasi metoda perhitungan dengan reduksi digit tak signifikan tersebut.

V. Daftar Pustaka

[1] David P. Anderson, *Public Computing: Reconnecting People to Science*, Space

Sciences Laboratory University of California - Berkeley, March 21, 2004.

[2] Glazunov, Nik M., Development of Quality Interval, Algorithms Software and Systems, *A Supplement to the International Journal of Reliable Computing*, APIC 1995.

[3] John Langford, Tutorial on Practical Prediction Theory for Classification, *Journal of Machine Learning Research* 6, 273–306, 2005.

[4] Kreinovich. V, Data Processing Beyond Traditional Statistics : Applications of Interval Computations. A Brief Introduction, *A Supplement to the International Journal of Reliable Computing*, APIC, 1995.

[5] Lim C.C, Zhao W., Performance Analysis of Dynamic Multitasking Imprecise Computation System, *IEEE PROCEEDINGS*, Vol. 138, no 5, September 1991.

[6] Lin Rong and Schwing James L., *A Non-Binary Parallel Arithmetic Architecture*, Department of Computer Science, SUNY at Geneseo Geneseo, NY 14454 Department of Computer Science, Central Washington University Ellensburg, WA 98926.