

IMPLEMENTASI METODA PRE-PROCESSING UNTUK MENINGKATKAN LAJU AKURASI HASIL KOMPUTASI OPERASI PENJUMLAHAN SEKUENSIAL MULTI OPERAND SISTEM WAKTU NYATA

Oleh :

Andrizal

Staf Pengajar Jurusan Teknik Elektro Politeknik Negeri Padang

Telp 0751-72590 Fax 0752-72576 E-mail : andrizal@polinpdg.ac.id

ABSTRACT

Real-time system works in a limited period of time in accordance with the deadline time given. The success of a real time system performance is determined by several time components such as release time, processing time and time deadlines. Problems arise when the processing time is not enough due to the delay factor deadline or other factors, while the deadline time has come and the process had to be stopped. When the process is stopped, the results obtained have not been enough to be used as the end result of data systems fail to meet the given time constraint. There are several ways to overcome this, one is to perform preprocessing before the end of the process is done. In the sequential preprocessing system multioperand summation of the operand-operand which can be summed sort operand-operand which has significant influence on the results done first. When the process had to be stopped due to a deadline is coming then the accuracy of the results produced may be used as the final result.

This study utilized the method as a preprocessing ascend in sorting systems to sort operand-operand values to be summed from largest to smallest. The results show the accuracy of the rate of change of the summation process in accordance with the needs of real-time system when the process had to be stopped before the end.

Keywords : Real-Time, multi operand adder, pre processing.

PENDAHULUAN

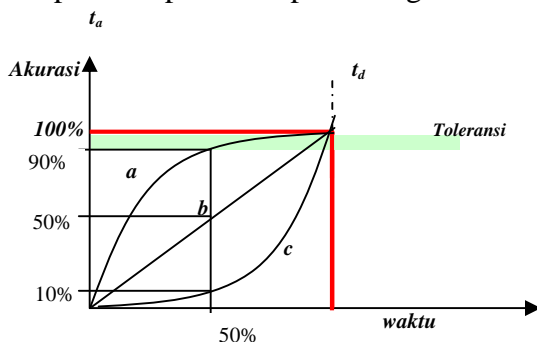
Sistem waktu-nyata merupakan sistem komputer yang berkaitan erat dengan peryaratan waktu-tenggat (deadline). Sistem ini memerlukan proses yang cepat, respons tepat waktu dan akurasi hasil yang sebaik-baiknya [Alan 2002, Bruce 1999]. Sistem dikatakan berhasil jika response time yang dihasilkan masih dalam batas nilai toleransi yang diterima dan dalam waktu tidak melampaui Time deadline yang diberikan.

Banyak usaha yang dilakukan dalam memenuhi kriteria proses sistem waktu-nyata antara lain menggunakan teknik: scheduling, parallelism, preprocessing, data reduction, data compression.

Usaha-usaha diatas mengacu kepada bagaiman tersedianya hasil dari proses saat dibutuhkan sesuai dengan deadline dan dengan pencapaian nilai akurasi hasil yang berada dalam nilai yang bisa ditoleransi.

Bentuk umum dari kurva akurasi hasil komputasi terhadap waktu proses dapat dilihat pada gambar 1 [Kuspriyanto, Kerlooza YY, 2004].

Untuk sistem waktu nyata, bentuk kurva laju akurasi yang diharapkan adalah seperti pada gambar 1(a). Hal ini diperlukan karena sifat-sifat waktu parameter proses sistem yang diolah yaitu waktu kedatangan (t_a) proses, lama proses (T_p) dan waktu-tenggat T_d dapat diprediksi pada saat perancangan sistem.



Gambar 1 : Bentuk kurva akurasi menurut cara proses (a). Proses dimulai dengan data yang besar pengaruhnya (significant) terhadap hasil (b). Proses dimulai dengan data yang linear pengaruhnya terhadap hasil (c). Proses dimulai dengan data yang kecil pengaruhnya (significant) terhadap hasil

Jika proses mengalami gangguan saat sistem harus melakukan respon terhadap suatu kejadian dalam waktu yang terbatas, seperti terjadinya keterlambatan kedatangan proses (t_a), atau waktu-tenggat T_d dipercepat. Respon waktu yang memungkinkan memenuhi syarat kriteria real time adalah respon yang terlihat pada gambar 1(a). Hal ini dimungkinkan karena perubahan laju akurasi proses significant sesuai dengan perubahan waktu proses. Pada area waktu proses 50% laju akurasi hasil komputasi yang dicapai sekitar 90%. Dengan kata lain jumlah akurasi proses komputasi yang sedemikian telah memungkinkan sistem untuk mengambil keputusan, dibandingkan dengan kurva response gambar 1 (b) dan 1 (c) dengan nilai akurasi proses masing-masing 50% dan 10 %.

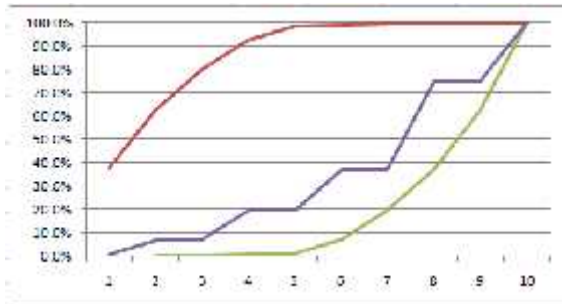
Pada proses penjumlahan multi-operand, nilai akurasi hasil proses yang dilakukan dalam suatu waktu tergantung kepada cara atau algoritma proses yang digunakan. Secara ideal proses yang digunakan tidak akan berpengaruh terhadap kinerja, jika *deadline* sudah datang, hasil yang diinginkan sudah tersedia dan siap untuk digunakan dengan akurasi terbaik.

Pada operasi penjumlahan dua operand terdapat beberapa algoritma seperti Carry Save Adder, Row Adder Tree, Ripley Carry Adder dan lain-lain.

Ketika jumlah operand yang akan dijumlahkan lebih dari dua atau multioperand, proses dapat dilakukan secara sekuensial atau berurutan maupun secara parallel. Pada penjumlahan multioperand sekuensial adalah dengan proses menjumlahkan data berurutan, sesuai dengan posisi element data. Sedangkan dengan proses parallel dibutuhkan perangkat keras pendukung yang mampu melakukan operasi parallel. Unit penjumlahan multioperand sekuensial secara umum memiliki laju perubahan nilai akurasi tergantung dengan nilai-nilai operand yang dijumlahkan. Ketika nilai operand-operand diurutan awal dan seterusnya bernilai besar, dan seterusnya bernilai kecil seperti terurut dari besar kekecil, maka dimungkinkan mendapatkan nilai laju akurasi seperti pada gambar 1 (a). Tetapi jika nilai operand-operand seperti terurut dari nilai yang kecil ke yang besar, laju akurasi yang dihasilkan seperti pada gambar 1(c). Dan jika besar nilai operand-operand acak, maka kemungkinan akan mendapatkan nilai laju akurasi seperti pada gambar 1(b).

Tabel 1 memperlihatkan tingkat laju akurasi penjumlahan multioperand dalam 3 variasi posisi nilai-nilai operand. Kurva 2(a) memperlihatkan proses penjumlahan dilakukan setelah

mengurutkan operand-operand dari nilai terbesar ke terkecil. Pada kurva 2(b) posisi nilai operand-operand secara acak dan kurva 2(c) dilakukan pengurutan nilai kecil ke besar.



Gambar 2. Persentasi laju akurasi hasil proses penjumlahan

berdasarkan posisi nilai operand
a. Dengan mengurutkan operand dari bernilai besar ke kecil. b. Dengan mengurutkan operand dari nilai kecil ke besar. c. Posisi Nilai operand acak

Laju akurasi kurva gambar 2(b) akan meningkat sesuai dengan itrasi proses, peningkatan persentasi laju akurasi cukup significant karena dilakukan pre-processing yaitu dengan menambahkan

proses sorting dari besar ke kecil sebelum proses komputasi.

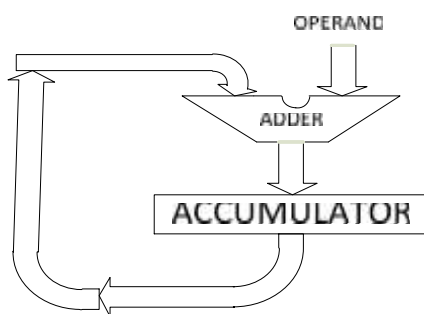
Berdasarkan laju akurasi yang didapat, maka dimungkinkan algoritma ini diterapkan pada aplikasi waktu nyata. Ada pun cara yang paling tepat digunakan adalah dengan cara mengolah data yang besar pengaruhnya terhadap hasil lebih dulu sehingga nilai akurasi yang didapatkan sudah berada pada batas nilai toleransi walaupun proses terpaksa dihentikan atau waktu proses sudah habis.

Penelitian ini bertujuan untuk menganalisa kemampuan perubahan nilai akurasi pada multioperand adder sequential dengan menambahkan preprocessing sorting data.

Rancangan ini masih memiliki batasan yaitu hanya digunakan pada bilangan tidak bertanda. Hasil yang diharapkan adalah dengan adanya preprocessing, didapatkan hasil nilai antara dengan perubahan akurasi yang sesuai dengan kebutuhan penjumlahan multioperand aplikasi waktu nyata.

Tabel 1. Laju akurasi hasil penjumlahan multioperand

Iterasi	Posisi Operand Terurut Dari Besar Ke Kecil			Posisi Operand Terurut Dari Kecil Kebesar			Posisi Acak		
	Operand	Hasil	Akurasi	Operand	Hasil	Akurasi	Operand	Hasil	Akurasi
1	15000	15000	37.6%	5	5	0.0%	200	200	0.5%
2	10000	25000	62.7%	20	25	0.1%	2456	2656	6.7%
3	7000	32000	80.2%	100	125	0.3%	123	2779	7.0%
4	5000	37000	92.7%	123	248	0.6%	5000	7779	19.5%
5	2456	39456	98.9%	200	448	1.1%	20	7799	19.5%
6	200	39656	99.4%	2456	2904	7.3%	7000	14799	37.1%
7	123	39779	99.7%	5000	7904	19.8%	5	14804	37.1%
8	100	39879	99.9%	7000	14904	37.3%	15000	29804	74.7%
9	20	39899	100.0%	10000	24904	62.4%	100	29904	74.9%
10	5	39904	100.0%	15000	39904	100.0%	10000	39904	100.0%



Gambar 2. Arsitektur sequential multioperand adder konvensional.

Unit penjumlahan multioperand sekuensial memulai proses penjumlahan secara acak sesuai dengan operand yang masuk ke akumulator lebih dulu. Arsitektur rangkaian terlihat pada Gambar 2. Keluaran dari akumulator dijumlahkan kembali dengan operand berikutnya, proses ini berulang sampai semua operand selesai diproses. Hasil akhir disimpan pada akumulator. Algoritma proses, ini memulai operasi secara acak.

```
for (i=0; i<=n; i++)
{
    Acc = Acc + Op [i]
}
```

METODOLOGI PENELITIAN

Waktu proses perhitungan (T_p) dalam sistem komputer dapat dibuat fleksibel (diperkecil) melalui penggunaan unit aritmatika yang dapat dihentikan prosesnya setiap saat dan memiliki *intermediate result* yang dapat diambil dan digunakan. Hasil yang didapat haruslah memiliki nilai akurasi yang terbaik. Hal ini dapat dilakukan dengan cara mengolah data yang terpenting lebih dahulu dengan cara menambahkan preprocessing, agar hasilnya memiliki nilai akurasi yang optimal disaat awal. Dengan demikian rancangan system perkalian waktu nyata harus memiliki sifat-sifat :

- Data yang *significant* (yang mengakibatkan hasil perhitungan mendekati hasil akhir) harus diproses lebih dulu.
- Dengan lebih dulu mengolah data-data yang *significant*, pada umumnya akan mempercepat perolehan hasil akhir atau mempersingkat waktu rata-rata proses (T_p).

Konsep dengan cara mendahulukan pengolahan terhadap data-data yang memiliki pengaruh *significant* terhadap hasil akhir inilah yang digunakan sebagai pendekatan dalam penelitian ini. Untuk mendapatkan data yang *significant* pengaruhnya terhadap hasil, dilakukan dengan menggunakan metoda *sorting* sebagai preprocessing. Akhirnya diharapkan ketika terjadi perubahan lingkungan yang menyebabkan proses terpaksa dihentikan atau *deadline* proses sudah datang maka akan memberikan nilai hasil proses yang masih berada dalam batas toleransi. Sistem semacam ini tentu bermanfaat untuk digunakan dalam proses *real-time*.

HASIL PENELITIAN

Konsep dasar preprocessing multioperand adder adalah dengan memproses operand-operand yang berpengaruh *significant* terhadap akurasi hasil menggunakan metoda *sorting*. Hasil yang didapatkan adalah laju nilai akurasi yang sesuai dengan kebutuhan sistem waktu nyata. Untuk Uji coba dan simulasi rancangan sistem digunakan perangkat lunak C++ Builder, sehingga dapat dianalisis kemampuan algoritma ini.

Rancangan arsitektur ini diperlihatkan pada Gambar 3. Semua operand ditampung sementara pada buffer sebelum dilakukan *sorting*. Hasil *sorting* ini dijumlahkan diakumulator, dan hasil penjumlahan akumulator ini dijumlahkan

kembali dengan operand berikutnya sampai operand terakhir.

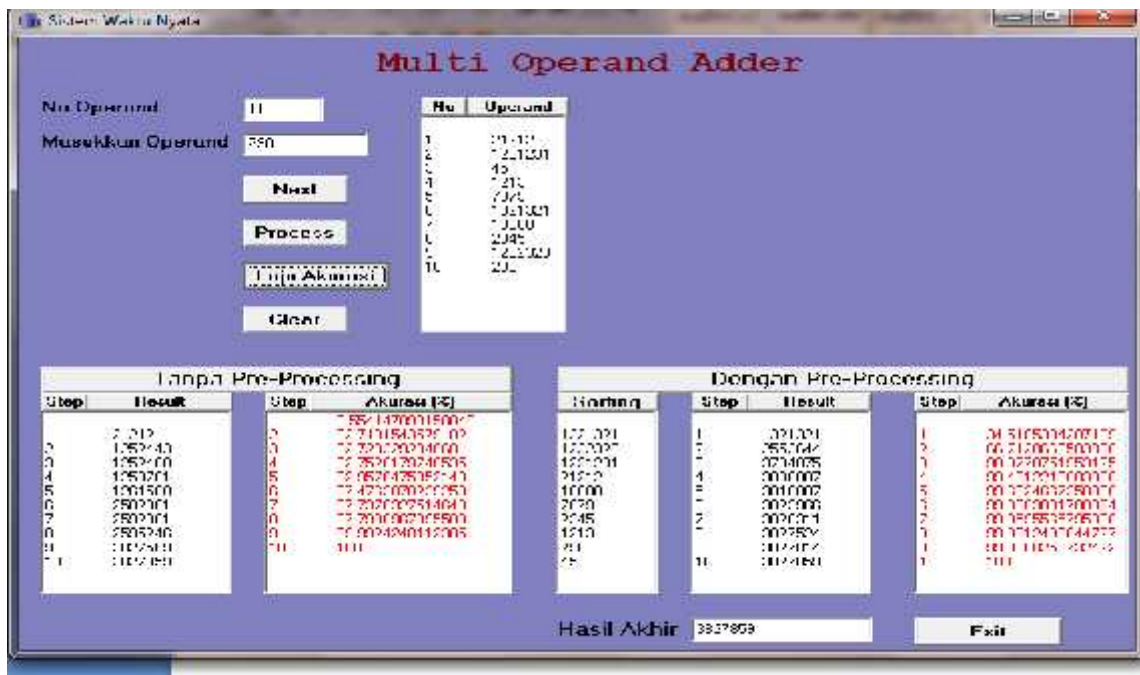
Tabel 2. Hasil teoritis laju nilai akurasi data acak metoda multi operand adder konvensional sejumlah penjumlahan terhadap sejumlah data acak.

	Data	Hasil Penjumlahan	Akurasi (%)
1	143567	0	0
2	100000	143567	52.967
3	20000	243567	89.861
4	4000	263567	97.240
5	2345	267567	98.716
6	1000	269917	99.581
7	280	270912	99.950
8	90	271192	99.970
9	35	271282	99.990
10	10	271317	100.000

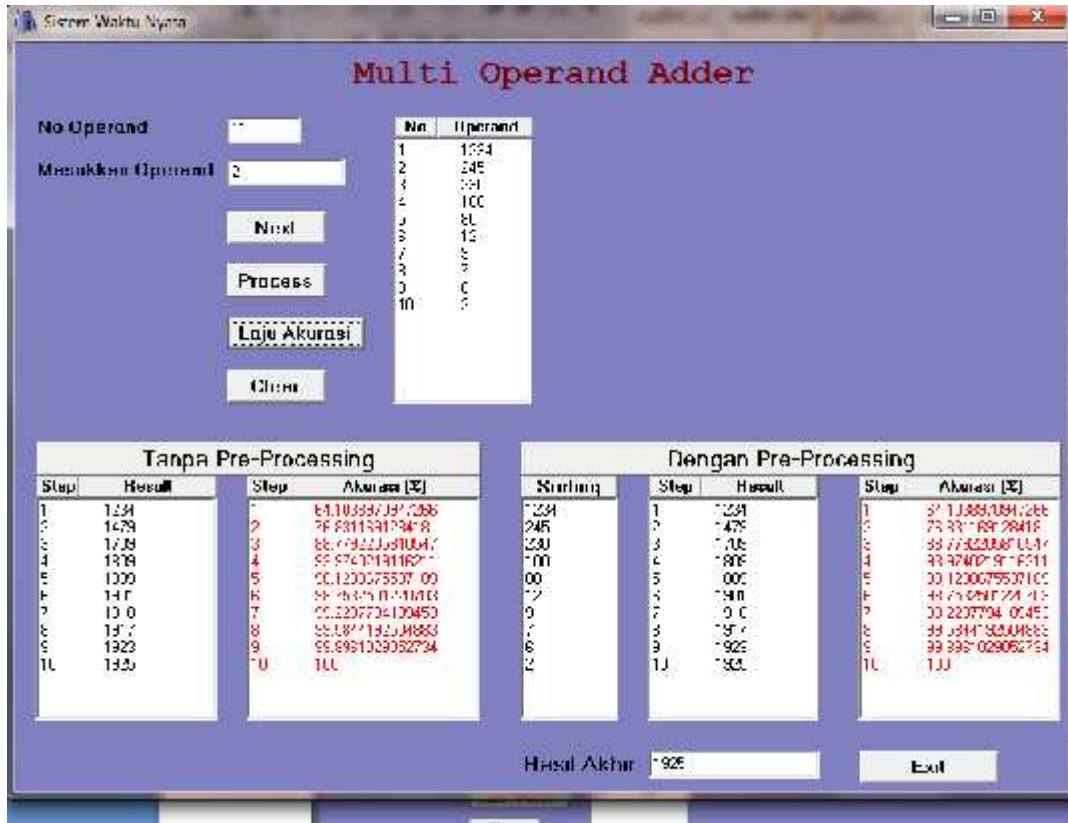
Tabel 3. Hasil teoritis data yang samalaju nilai akurasi data acak metoda preprocessing multi operand adder

	Data	Hasil Penjumlahan	Akurasi (%)
1	1000	0	0
2	20000	1000	0.368
3	10	21000	7.747
4	4000	21010	7.751
5	100000	25010	9.227
6	35	125010	46.121
7	2345	125045	46.134
8	143567	127390	46.999
9	90	270957	99.966
10	280	271047	100

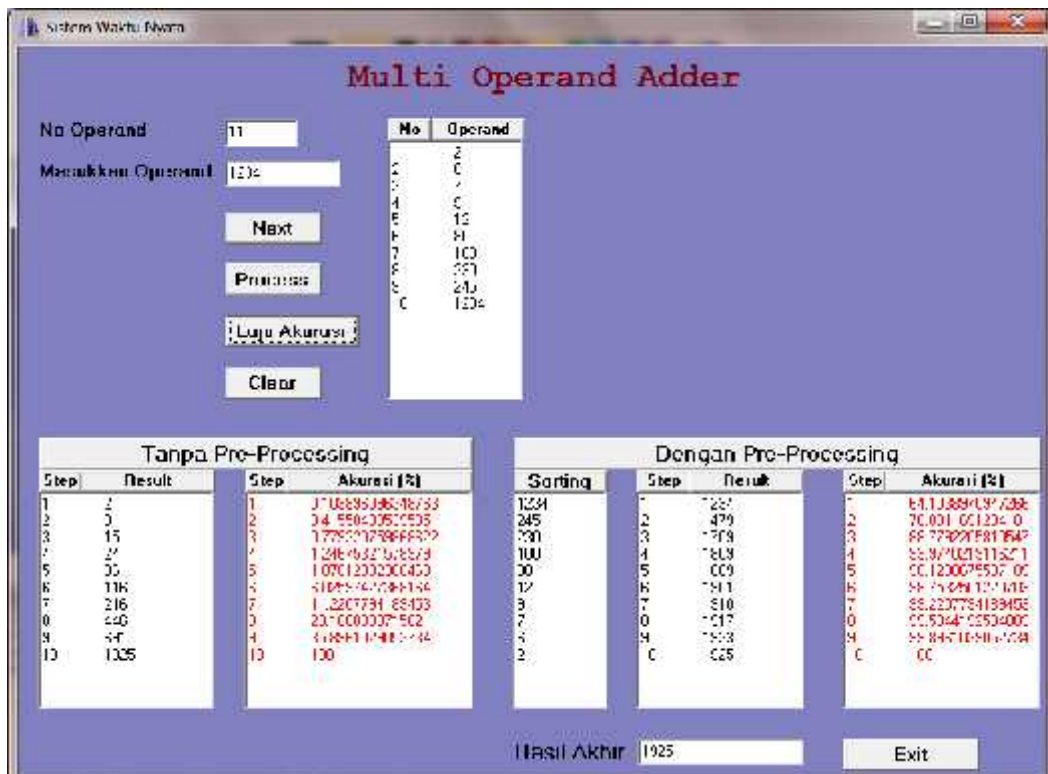
Hasil pengamatan beberapa operasi penjumlahan multioperand terhadap sejumlah data diperlihatkan pada gambar 4.



Gambar 4a. Hasil ujicoba kedua algoritma



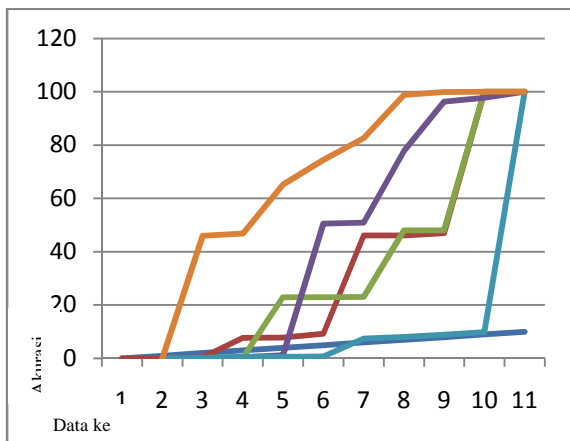
Gambar 4b. Hasil ujicoba terbaik algoritma konvensional



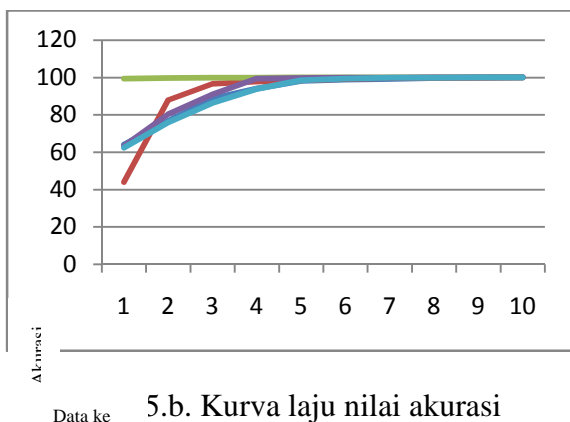
Gambar 4c. Hasil terburuk algoritma konvensional dan terbaik algoritma preprocessing

Dari simulasi terlihat bentuk laju nilai akurasi terhadap oprasi sejumlah operand yang dihasilkan dengan metoda preprocessing memiliki laju akurasi yang tinggi pada saat awal. Sebaliknya dengan metoda konvensional, laju nilai akurasi saat awal rendah, dan baru meningkat pada akhir proses. Ini berarti metoda preprocessing memperoleh nilai laju akurasi yang lebih baik dan sesuai dengan kebutuhan waktu nyata.

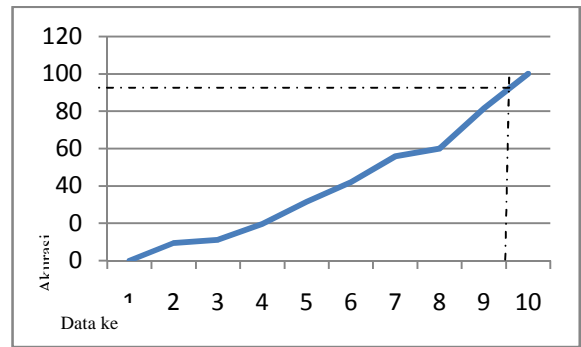
Selanjutnya ditunjukkan hasil pengujian laju akurasi terhadap sejumlah data acak. Bentuk kurva laju nilai akurasi terhadap iterasi terlihat seperti Gambar 5 a dan b.



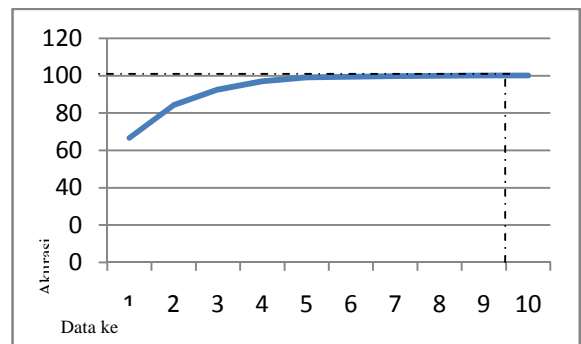
Gambar 5.a. Kurva laju nilai akurasi data hasil akhir sejumlah penjumlahan multioperand adder metoda konvensional



Gambar 5.b. Kurva laju nilai akurasi data hasil akhir sejumlah penjumlahan multioperand adder metoda preprocessing.



Gambar 6.a. Rata-rata kurva laju nilai akurasi penjumlahan multioperand adder metoda konvensional



Gambar 6.b. Rata-rata kurva laju nilai akurasi penjumlahan multioperand adder metoda preprocessing

Jika diamati hasil proses yang telah diperoleh untuk masing-masing algoritma, jelas terlihat bahwa algoritma penjumlahan multioperand metoda preprocessing selalu memiliki nilai akurasi yang lebih baik dari algoritma metoda konvensional.

Jika pada rentang proses penjumlahan terpaksa dihentikan, maka nilai akurasi respon sistem secara keseluruhan untuk metoda preprocessing lebih baik dibandingkan jika sistem menggunakan algoritmanya preprocessing (konvensional). Untuk lebih jelasnya dapat dilihat pada contoh respon laju nilai akurasi pada Gambar 6 a dan b pada garis putus-putus memperlihatkan laju akurasi yang didapat 100% untuk

metode *preprocessing* dan 96% untuk metode konvensional.

Jadi *intermediate result* yang dihasilkan dapat memenuhi kriteria untuk penggunaan perkalian waktu nyata.

KESIMPULAN

Berdasarkan data dan analisis yang diperoleh pada penelitian ini, dapat diambil kesimpulan sebagai berikut :

1. Dengan *preprocessing ascending sorting* seluruh operand terlebih dahulu, dapat memperbaiki laju nilai akurasi hasil proses, sehingga hasil proses yang didapat memiliki nilai akurasi yang tinggi pada iterasi-iterasi awal.
2. Jika proses terpaksa dihentikan sebelum berakhir, nilai akurasi yang didapat lebih baik dibandingkan jika proses yang tanpa *preprocessing*.
3. Hasil penelitian ini sangat menguntungkan peningkatan kinerja sistem waktu-nyata, yaitu tersedianya hasil antara proses (*intermediate result*) yang dapat diakses setiap saat dengan nilai akurasi yang terbaik jika waktu proses terpaksa diperpendek atau dihentikan.

DAFTAR PUSTAKA

- Andrizal. 2010, *Analisis Keandalan Algoritma Perkalian Braun Array Multiplier Untuk Operasi Perkalian Waktu Nyata*, Jurnal Elektron Jurusan Teknik Elektro Politeknik Negeri Padang.
- Andrizal.2009, *Keandalan metoda MSB-P1 algoritma Shift-And-ADD Pada Operasi Perkalian waktu nyata*, Poli Rekayasa, Volume 4.
- Andrizal. 2005, *Perancangan unit aritmatika perkalian waktu nyata*

algoritma Shift-And-ADD metoda MSB-P1, tesis, ITB.

- Alan C. Shaw.2002, *Real-Time Systems and Software*, John Wiley & Sons.
- Bruce powel Douglass 1999, *Doing Hard Time: Developing Real-Time Systems with UML, Objects, Frameworks, and Patterns*, Addison Wesley (1999).
- Kuspriyanto, Kerlooza, Y.Y.2004. *Toward New Real-Time Processor : The Multioperand MSB-First Real-Time Adder*, Proceedings of DSD'2004 Euromicro Symposium on Digital System Design, Rennes-France, IEEE Computer Society.
- Kuspriyanto, Kerlooza, Y.Y.2004 *Keandalan Unit Multioperand MSB-First Realtime Adder Pada Operasi Penjumlahan Data Acak*, Proceeding SITIA 2004, Seminar on Intelligent Technology and Its Applications, Surabaya, May 18th.
- Levi, S.T., Agrawala, A.K.1990. *Real-Time System Design*, McGraw-Hill.
- Winarno S, Jazi Eko I, Khabib Mustofa, 2010 *Arsitektur Real Time System Sebagai Pemantau Jaminan QOS*. Jurnal informatika Vol 6 No 2 Des 2010. Universitas Maranatha.