

## IMPLEMENTASI INSTRUMENTASI MAYA UNTUK PEMBANGKITAN SINYAL UJI RANGKAIAN PENGHITUNG N-16 TERKOREKSI SPR RSG-GAS

Edison, Sujarwono, Ranji Gusman, Purwadi  
Pusat Reaktor SerbaGuna-BATAN

### ABSTRAK

Untuk mengatasi pengaruh penuaan pada sistem proteksi reaktor, PRSG mulai merencanakan pembaharuan (*refurbishment*) sistem tersebut. Sesuai dengan rekomendasi ahli IAEA, jika PRSG mengganti dengan sistem berdasarkan digital maka sebaiknya dibuat simulator untuk memastikan bahwa *programmable logic controller* industri yang baru memenuhi kinerja bagian sistem yang bersesuaian. Salah satu kegiatan untuk melaksanakan rekomendasi ini adalah pembuatan simulator rangkaian penghitung N-16 terkoreksi yang merupakan bagian dari sistem. Untuk melakukan karakterisasi dengan simulator ini diperlukan sinyal uji. Karena generator sinyal uji analog yang tersedia tidak dapat digunakan lagi, maka dibuat generator sinyal uji yang diprogram menggunakan program LabVIEW. Panil depan generator sinyal uji yang dibangun pada layar komputer merupakan instrumen maya. Dengan demikian perangkat keras panil depan untuk antar muka dengan pengguna tidak perlu diadakan. Disamping itu panil depan yang dibuat dapat secara luwes disesuaikan dengan kebutuhan karakterisasi. Generator sinyal uji ini dapat diprogram untuk menghasilkan sinyal *step* dan *ramp* secara berurutan yang dapat diaktifkan pada kontrol panil depan. Hasil pengukuran dengan oskiloskop menunjukkan bahwa pola dan parameter sinyal output yang dihasilkan sesuai dengan instruksi pada panil depan dengan akurasi cukup untuk keperluan karakterisasi.

### ABSTRACT

*To mitigate the ageing effect to the reactor protection system, PRSG begin to plan refurbishment of the system. According to the recommendation of IAEA expert that if PRSG replace to digital based system then PRSG should construct a simulator to ensure that an industrial programmable logic controller satisfy the performance of the existing corresponding part. One of the activities to implement the recommendation is construction of a simulator for the N-16 corrected computing circuit which is a part of the system. The test signal is needed to characterize the existing computing circuits. Because the test signal is not available any more, a test signal generator which is programmed by using LabVIEW program is developed. The front panel of the test signal generator, which is built on the screen of a computer, is a virtual instrument, therefore the hardware of the panel for user interface has not to be purchased. Furthermore, the front panel built can be freely customized to support the characterization. This generator can be programmed to generate step and ramp signal, which can be sequentially activated at the front panel control. Result of test using a oscilloscope show that the pattern and the parameter of the output signals follow the instruction at the front panel. The accuracy of generated signals is good enough for the characterization.*

### PENDAHULUAN

Sistem Proteksi Reaktor (SPR) RSG-GAS sudah beroperasi bersamaan dengan komisioning reaktor. Dengan demikian akibat pengaruh penuaan beberapa komponen dari SPR sudah mengalami kegagalan bahkan untuk komponen yang sama ada yang lebih dari sekali gagal dan dilakukan penggantian. Akibat dari penggantian ini persediaan beberapa komponen sudah habis terpakai dan tidak ada pemasok lagi yang menyediakannya karena sudah tidak diproduksi lagi (sudah usang). Sesuai dengan ketentuan maka untuk menyelesaikan persoalan keusangan organisasi pengoperasi dalam hal ini PRSG mulai merencanakan pembaharuan (*refurbishment*) SPR. Mengingat bahwa SPR merupakan struktur, sistem dan komponen (SSK)

yang penting bagi keselamatan maka desain dan kualitas dari SPR baru yang direncanakan harus sungguh-sungguh mengikuti SPR terpasang.

Dalam fungsinya melakukan tindakan protektif untuk menyelamatkan reaktor, SPR harus handal sehingga standar kualitas dan prinsip-prinsip redundansi, diversitas dan gagal aman harus terimplementasi. Beberapa variabel proses yang mendeteksi kejadian diukur langsung dari besaran fisisnya dan dievaluasi apakah melebihi harga batas atau tidak. Namun demikian beberapa variabel yang mendeteksi kejadian atau kecelakaan jenis lain tidak berasal langsung dari akuisisi data parameter operasi reaktor tetapi melalui pemerosesan terlebih dahulu di bagian rangkaian penghitung SPR. Sesuai dengan salah satu rekomendasi ahli dari IAEA, PRSG sebaiknya membuat simulator digital terlebih dahulu

untuk menjamin bagian digital pengganti memenuhi kinerja bagian SPR analog terpasang saat ini.

Untuk menjamin rangkaian penghitung pengganti memenuhi kinerja rangkaian penghitung analog terpasang tidak cukup dengan mempelajari dokumen yang ada karena pada kenyataannya dokumen tersebut kurang lengkap atau berupa garis besar saja. Oleh karena itu perlu dilakukan karakterisasi bagian demi bagian dengan mensimulasikannya dan melakukan verifikasi terhadap kinerja rangkaian terpasang. Dalam tulisan ini akan dibahas mengenai pembuatan generator sinyal uji untuk rangkaian penghitung N-16 terkoreksi digital yang dibutuhkan untuk memastikan bahwa rangkaian tersebut memenuhi kinerja rangkaian terpasang. Generator sinyal uji ini dibuat dengan mengaplikasikan instrumentasi maya menggunakan program LabVIEW. Dengan instrumentasi maya ini panil depan untuk kontrol dan indikator maya dibuat pada layar komputer sehingga tidak ada biaya untuk perangkat keras komponen tersebut. Disamping itu keluwesan dalam input/output data dapat dengan mudah diimplementasikan dengan hanya memprogram saja sehingga dapat disesuaikan dengan kebutuhan pengguna.

## PERANCANGAN, PEMROGRAMAN DAN PERAKITAN ALAT

### Desain Dasar

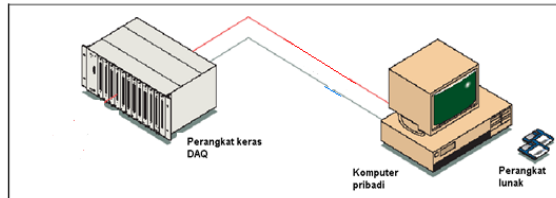
Berdasarkan dokumen perawatan dan dokumen rangkaian penghitung N-16 terkoreksi maka generator harus memenuhi kriteria desain sebagai berikut:

Generator dapat menghasilkan sinyal uji berbentuk sinyal *step* dan sinyal *ramp* dengan rentang 0-10 Volt. Pengguna dapat memasukkan pola sinyal *step* dan *ramp* sehingga generator menghasilkan beberapa sinyal *step* dan *ramp* dengan besar/durasi berbeda secara berturut-turut.

### Perancangan Alat

Untuk mengimplementasikan dasar desain tersebut dibutuhkan komputer PC yang sudah diinstalasi dengan perangkat lunak program aplikasi

LabVIEW dan *driver* untuk perangkat akuisisi data. Kemudian komputer juga dihubungkan dengan perangkat akuisisi data (DAQ) yang menghasilkan sinyal berupa tegangan dengan pola sesuai dengan perintah program yang dijalankan di PC seperti diperlihatkan dalam Gambar 1.



Gambar 1. Susunan perangkat generator sinyal uji

### Pemrograman

Pemrograman generator sinyal uji dilakukan menggunakan program LabVIEW yang disebut instrumen maya, karena operasi dan tampilannya merupakan imitasi dari instrumen fisik. Untuk memenuhi kinerja generator yang ditetapkan dalam desain dasar, seperti pada instrumen fisiknya akan dibuat imitasi panil depan. Program LabVIEW untuk generator sinyal ini dinamakan "Sinyal Uji Rangkaian Pengukur Maya". Bagian program sebagai tempat pembuatan kode sumber grafis yang mendefinisikan fungsionalitas program grafis disebut Diagram Blok. Alur pemrograman generator sinyal uji ini diperlihatkan oleh *flowchart* dalam Gambar 2.

### Panil depan

Pada panil depan terdapat kontrol tempat pengguna menginputkan : banyaknya urutan, waktu, jenis, banyak sampel sinyal per urutan sinyal dan laju sampel sinyal (Gambar 3). Disamping itu pada panil depan juga terdapat tombol untuk pengoperasian, serta terdapat monitor berupa *chart* yang memberikan gambaran visual sinyal uji yang dihasilkan, ditampilkan dan diujikan. Panil depan memuat kontrol dan indikator seperti pada Tabel 1 berikut ini :

Tabel 1. Kontrol dan Indikator pada Panil Depan

Jenis	Nama Antar Muka Pengguna	Kegunaan
Tombol Boolean	Tombol STOP	Menghentikan Program
	Tombol Mulai Uji	Memulai Pengujian
	2 Tombol SIMPAN	Menyimpan data uji atau membatalkan
Kontainer	2 Kontrol Tab	Untuk menempatkan matrik kontrol yang dapat dipilih untuk data fluks neutron atau N-16

Tabel 1. Lanjutan

Jenis	Nama Antar Muka Pengguna	Kegunaan
Kontrol Numerik	Laju sampel	Untuk menginputkan se-lang waktu <i>update</i> output DAQ
	Rentang output	Rentang keluaran DAQ
Matrik Kontrol 10 x4	2 matrik satu untuk setiap tab	Sebagai tempat untuk meletakkan dan menyimpan kontrol numerik data sinyal dan tombol geser untuk memilih jenis sinyal
Kontrol numerik pada setiap baris matrik	Kontrol String Urutan	Untuk menamakan urutan pola sinyal
	Kontrol Jumlah Titik	Menentukan jumlah sampel untuk setiap pola
	Kontrol numerik Step/Ramp	Untuk menentukan tegangan step atau laju penambahan/ pengurangan tegangan <i>ramp</i> pada output DAQ
Tombol Boolean	Kontrol Step/Ramp	Untuk memilih jenis sinyal output DAQ
Indikator <i>Chart</i>	Chart sinyal uji	Tempat untuk menam-pilkan sinyal uji secara grafis

**Diagram Blok**

Diagram blok terdiri dari:

**Terminal** : pada diagram blok, semua objek panil depan berupa kontrol nampak sebagai sebagai terminal data. Terminal adalah pangkalan masuk dan keluar yang menukar informasi antara panil depan dan diagram blok.

**Node** : Disamping itu pada blok diagram terdapat juga *node-node* yang memiliki *input/output* dan melaksanakan operasi ketika program berjalan.

**Wire** : merupakan jalan untuk mentranfer data diantara abjek-objek diagramblok. Struktur : merupakan representasi grafis dari *loops* dan *case statement* pemrograman teks. Penggunaan struktur pada diagram blok untuk mengulangi blok-blok kode dan untuk mengeksekusi kode-kode sesuai kondisi atau dalam urutan khusus.

Detail dari diagram blok program ini diperlihatkan dalam Gambar 4 dan Gambar 5 dengan penjelasan struktur-struktur penting dari diagram blok sebagai berikut :

1. *While loop* “Pembentukan sinyal uji fluks neutron”. Loop ini akan dieksekusi mulai dari N=0 sampai N-1 = jumlah elemen dari matrik dikurangi satu. Mula-mula ketika N=0, fungsi *Index Array* akan mengembalikan elemen pertama. Elemen ini berupa *cluster* empat elemen yang diisi oleh user yaitu : nomor urut, jumlah titik sampel, jenis sinyal *step* atau *ramp* dan level sinyal atau laju *ramp*.

Selanjutnya data *cluster* ini akan diinputkan melalui *wire* ke *case structure* “Pemilihan jenis sinyal uji fluks neutron” yang memilih struktur yang akan dieksekusi sesuai dengan nilai yang dihubungkan kepada *selector terminal*.

Jika nilai *case* = *true* maka *structure* untuk membuat sinyal *ramp* akan dieksekusi. Mula-mula data jumlah titik sampel diinputkan ke

*count terminalloop* “Membuat sampel sinyal uji ramp satu persatu”. Sedangkan laju ramp dan laju sampel diinputkan ke fungsi aritmatika untuk memperoleh nilai titik-titik sampel yang akan dihasilkan. Perubahan nilai dari satu sampel ke sampel berikutnya selama selang waktu yang berkorespondensi dengan laju sampel adalah:

$$\Delta V = \frac{\text{laju ramp} \left( \frac{mV}{\text{detik}} \right)}{(1000 \frac{mV}{V})} = \frac{(\text{laju ramp}/1000)}{\text{laju sampel}} (V)$$

Nilai sampel untuk setiap iterasi *i* mulai dari nilai *i*= 0 adalah :

$$V_i = V_{i-1} + \Delta V,$$

dengan nilai  $V_{-1} = 0$  atau sama dengan nilai sebelumnya yang disimpan dalam register. Semua data sampel ini dihubungkan kepada *loop tunnel* sehingga terbentuk matriks dari sampel sinyal uji ini. Ketika *loop* “Membuat sampel sinyal uji ramp satu persatu” sudah dieksekusi N kali maka data matriks pada *loop tunnel*nya ditransfer ke fungsi *Build Array* yang akan membangun matrik dengan menambahkan matrik ini ke baris matrik sebelumnya. Akhirnya matrik ini dihubungkan ke register pada terminal output *While loop* “Pembentukan sinyal uji fluks neutron” untuk menyampaikan matriks hasil kepada iterasi berikutnya.

Jika nilai *case* = *false* maka *structure* untuk membuat sinyal *step* akan dieksekusi. Mula-mula

data jumlah titik sampel diinputkan ke *count terminalloop* "Membuat sampel sinyal uji step satu persatu". Berdeda dengan perlakuan pada sinyal *ramp*, hanya nilai sampel sinyal *step* yang diinputkan ke *loop*.

Semua data sampel ini dihubungkan kepada *loop tunnel* sehingga terbentuk matriks dari sampel sinyal uji ini. Ketika *loop* "Membuat sampel sinyal uji step satu persatu" sudah dieksekusi N kali maka data matriks pada *loop tunnel*nya ditransfer ke fungsi *Build Array* yang akan membangun matrik dengan menambahkan matrik ini ke baris matrik sebelumnya. Akhirnya matrik ini dihubungkan ke register pada terminal output *While loop* "Pembentukan sinyal uji fluks neutron" untuk menyampaikan matriks hasil kepada iterasi berikutnya.

Hal yang sama berlaku untuk pembentukan baris sinyal uji N-16.

2. *While loop* "Penampilan dan generasi sinyal uji". *Loop* ini dieksekusi N kali yaitu sejumlah seluruh sampel yang ada di masing-masing

kedua matriks. Setiap kali eksekusi isi masing-masing matrik dikembalikan elemen demi elemen sesuai nilai indeks pada fungsi *Index Array*. Kemudian elemen kedua matrik *bundle* untuk ditampilkan pada *chart* "Chart Sinyal Uji". Selain dari pada itu, masing-masing dari kedua elemen diinputkan kepada input DAQ. Kemudian DAQ menggenerasi sinyal output pada terminal outputnya dengan level sesuai dengan nilai sampel untuk setiap selang waktu  $t = 1000 / \text{laju sampel (mili detik)}$ .

## HASIL DAN PEMBAHASAN

Disini akan dibahas sinyal yang dihasilkan oleh generator yang diukur oleh perangkat DAQ USB 6009 seperti dalam Gambar 6. Sinyal ideal adalah sinyal yang sesuai dengan data input seperti pada Gambar 3. Data sinyal dalam percobaan ini diperlihatkan dalam Tabel 2 dibawah ini :

**Tabel 2.** Data Hasil Pengukuran Sinyal Uji

Sinyal ramp ideal					Sinyal ramp terukur				
Urutan	Durasi Δt (detik)	Perubahan nilai (V)		Laju Ramp (mV/detik)	Urutan	Durasi Δt (detik)	Perubahan nilai (V)		Laju Ramp (mV/detik)
		Awal	Akhir				Awal	Akhir	
I	40	5	5	0	I	39,9	4,99	4,99	0,00
II	83	5	3,498	-18,1	II	83,3	4,99	3,48	-18,13
III	286	3,498	9,904	22,4	III	285,1	3,48	9,88	-22,52
IV	270	9,904	9,904	0	IV	269,9	9,88	9,88	0,00

Sinyal step ideal			Sinyal step terukur		
Urutan	Durasi Δt (detik)	Nilai sinyal (V)	Urutan	Durasi Δt (detik)	Nilai sinyal (V)
I	40	5	I	39,9	4,95
II	83	5	II	83,3	4,95
III	286	5	III	285,1	4,95
IV	270	5	IV	269,9	4,95

Seperti diperlihatkan dalam Tabel 2., durasi sinyal terukur menggunakan perangkat DAQ USB 6009 menyimpang dari sinyal ideal. Penyimpangan terbesar sekitar 0,9 detik. Hal ini terjadi karena *timer* yang digunakan baik pada generator maupun pengukur adalah *timer software* yang bekerja berdasarkan *clock* sistem komputer. *Timer* seperti ini memang kurang akurat karena bergantung pada lama waktu eksekusi instruksi. Disamping itu *scan rate*

sebesar 10 sampel perdetik juga kurang tepat dalam mendeteksi awal dan akhir sinyal. Hal ini tidak dapat dihindari karena dari hasil percobaan *scan rate* sebesar 10 sampel perdetik merupakan kemampuan maksimal DAQ ini agar selang waktu pembacaan sinyal masih benar.

Kesalahan juga terjadi pada hasil pengukuran level sinyal yang berakibat pada laju *ramp*. Kesalahan terbesar adalah sebesar 18 mV.

Kesalahan ini berasal dari akurasi DAQ baik dalam generasi maupun dalam akuisisi data sinyal. Dari spesifikasi teknis diketahui bahwa akurasi DAQ sebesar 7 mV, dengan demikian perbedaan tegangan antara sinyal ideal dan terukur sebesar 18 mV merupakan konsekuensi wajar dari sistem percobaan ini.

Meskipun sinyal keluaran generator uji tidak tepat seperti sinyal idealnya namun untuk keperluan karakterisasi kinerja rangkaian N-16 terkoreksi sudah melebihi cukup karena pada dasarnya kita tidak melakukan pengukuran namun hanya mengetahui responnya secara kualitatif untuk pengembangan *software*. Untuk pengukuran atau pengujian sesungguhnya digunakan DAQ yang memiliki akurasi sinyal output dan input yang lebih tinggi dan memiliki *hardware timer* yang lebih akurat.

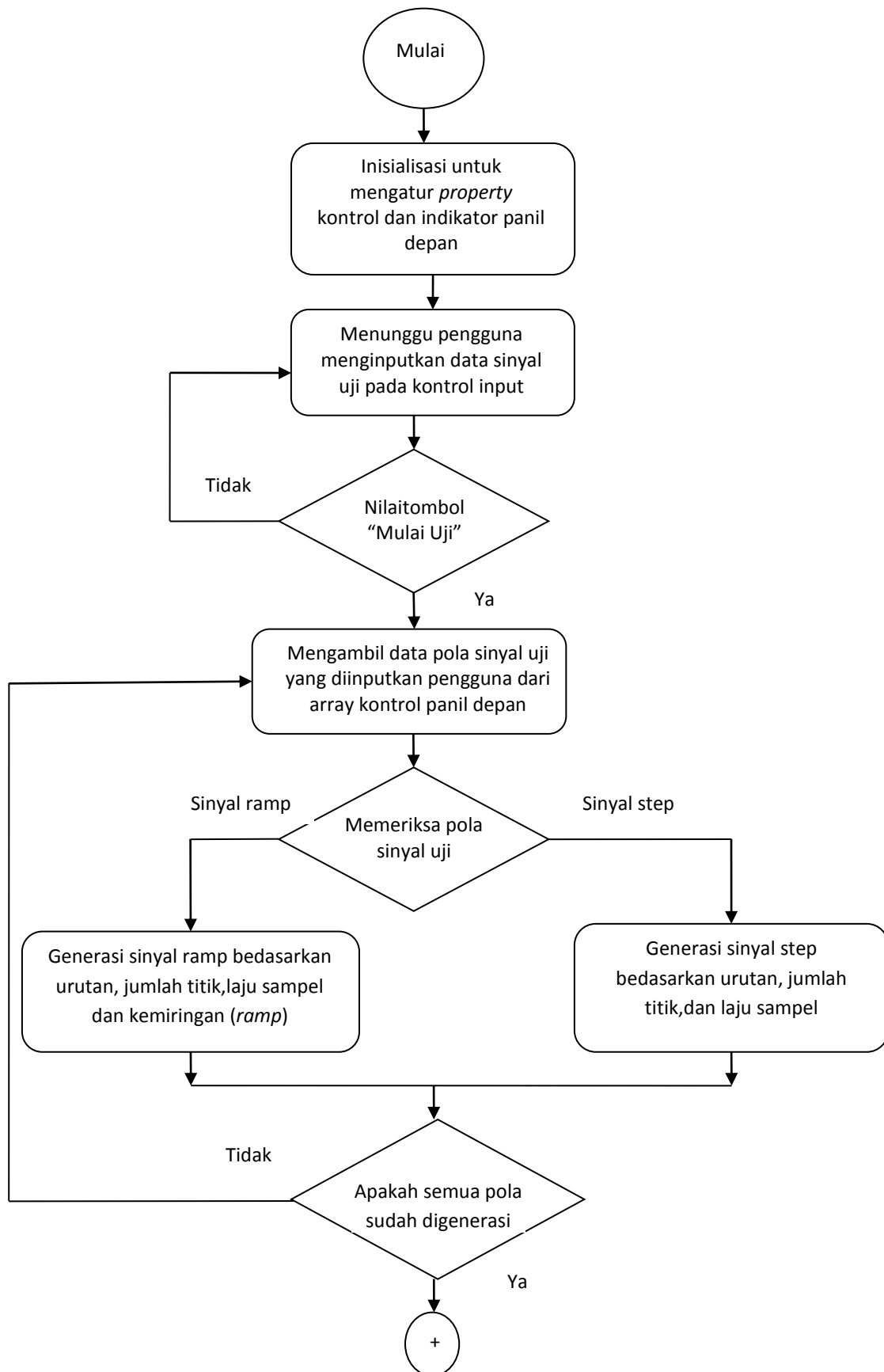
#### KESIMPULAN DAN SARAN

Kombinasi software dan konfigurasi hardware generator sinyal uji untuk rangkaian N-16 terkoreksi yang diimplementasikan dalam kegiatan ini cukup

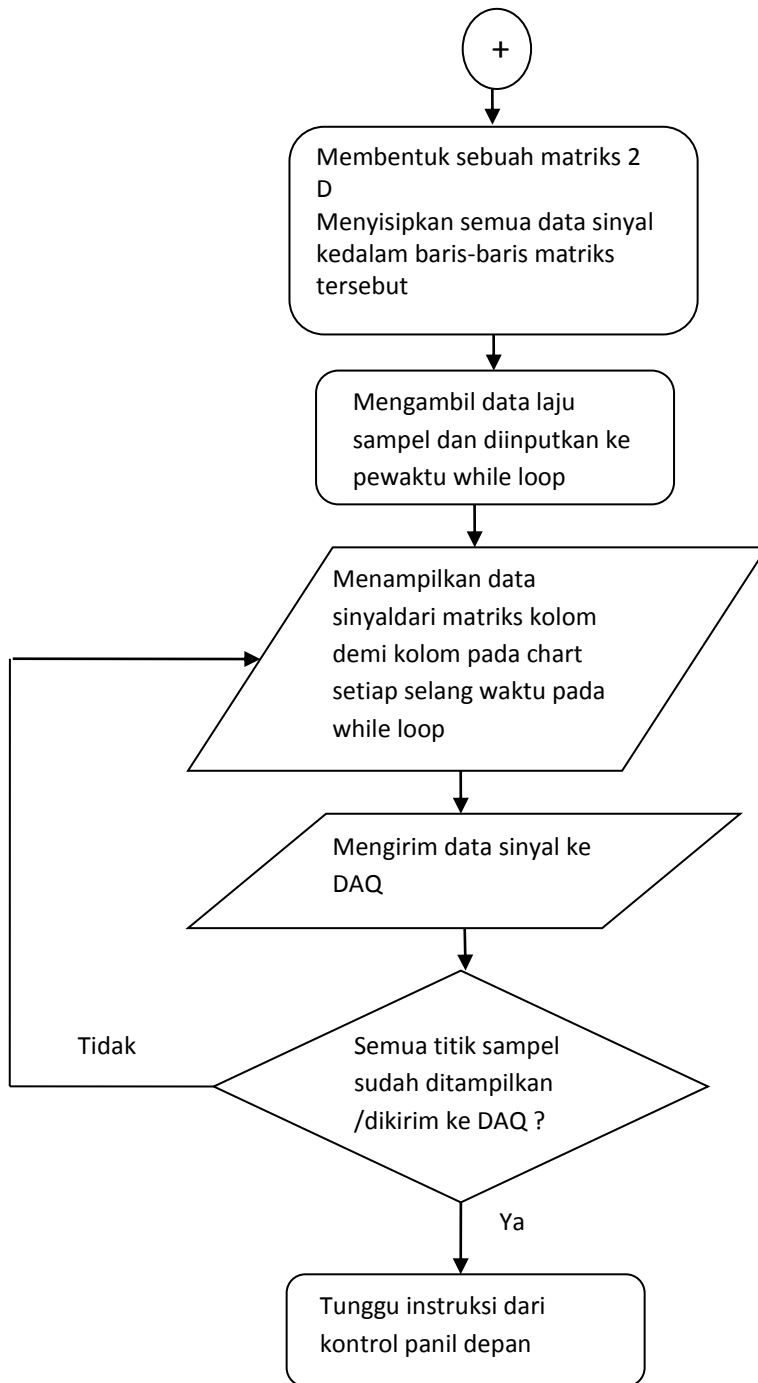
baik untuk mendukung pembuatan software. Untuk keperluan pengujian rangkaian riil yang membutuhkan ketelitian digunakan DAQ dengan spesifikasi akurasi output tegangan dan timer yang lebih tinggi.

#### DAFTAR PUSTAKA

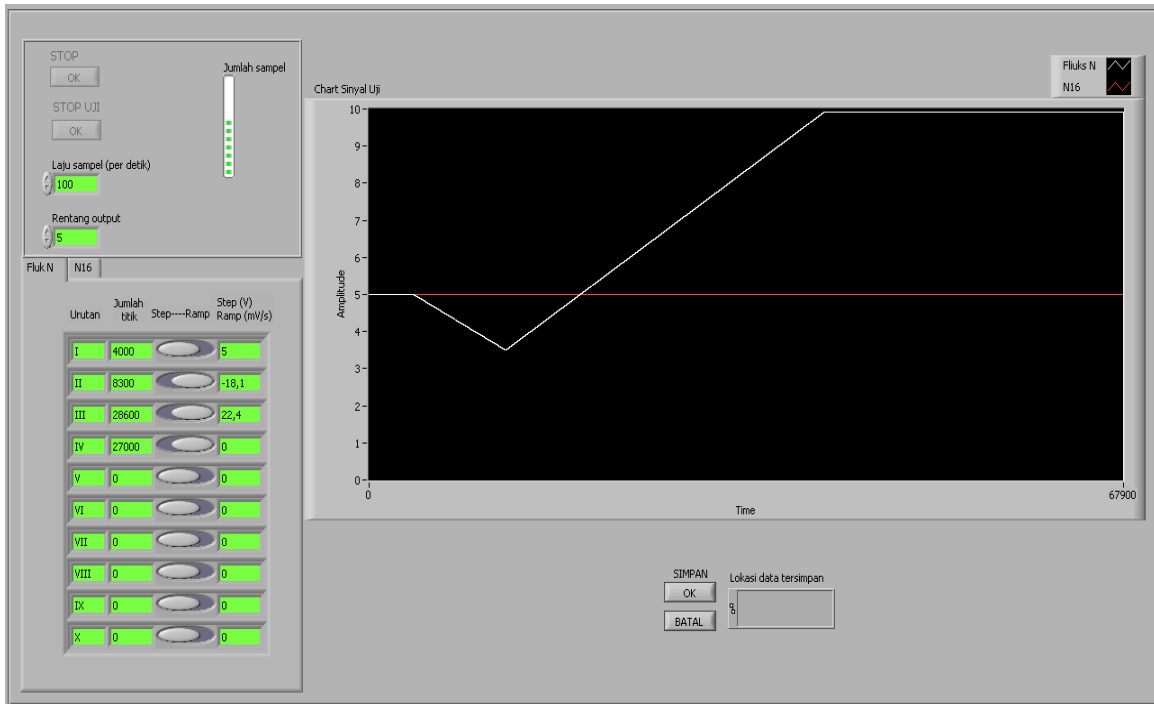
1. *TOP 9 Reactor Protection System, System Description*, Dokumentasi Pusat Reaktor Serba Guna, BATAN, 1987
2. *TOP 9 Reactor Protection System, Data Komissioning*, Dokumentasi Pusat Reaktor Serba Guna, BATAN, 1986
3. LabVIEW User Manual, NATIONAL INSTRUMENTS, 2003
4. LabVIEW Measurement Manual, NATIONAL INSTRUMENTS, 2003
5. User Guide and Specification, NI USB 6008/6009, NATIONAL INSTRUMENTS, 2012.
6. LabVIEW Help, NATIONAL INSTRUMENTS, 2004



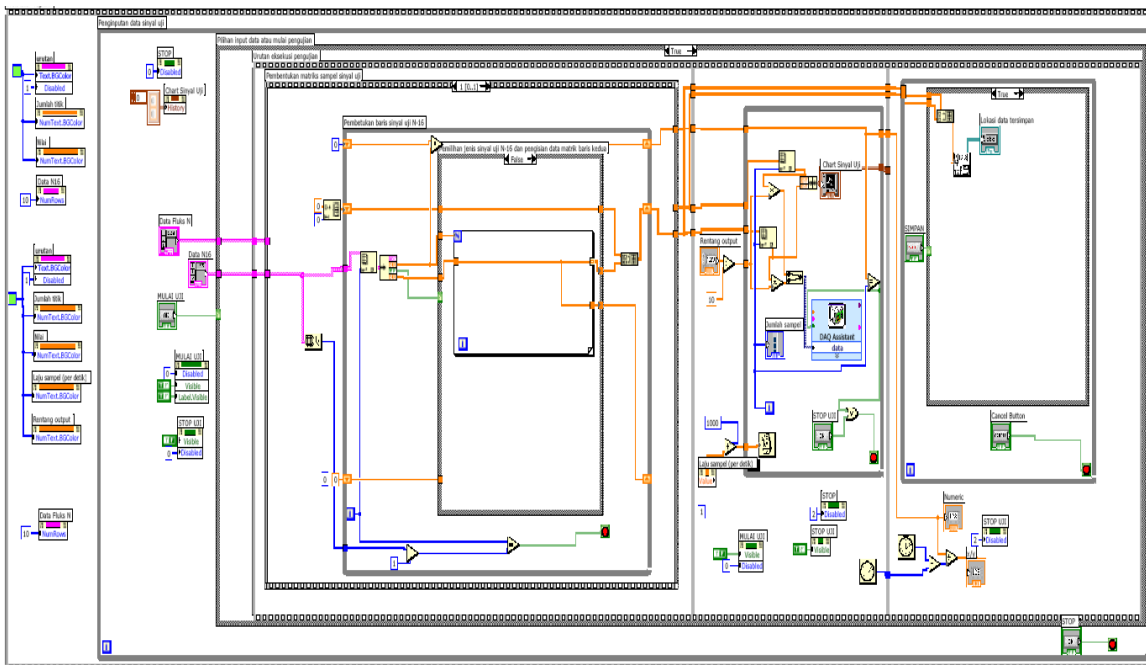
Gambar 2. Flowchart Pemrograman



Gambar 2. Flowchart Pemrograman (lanjutan)

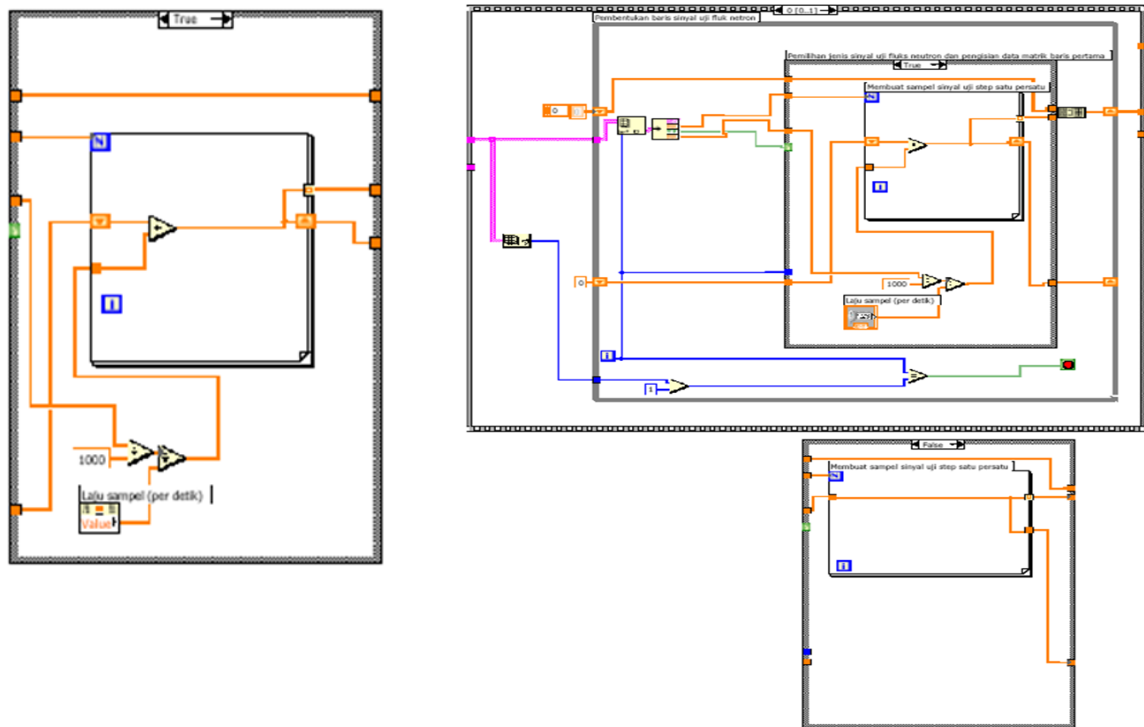


Gambar 3. Panil Depan

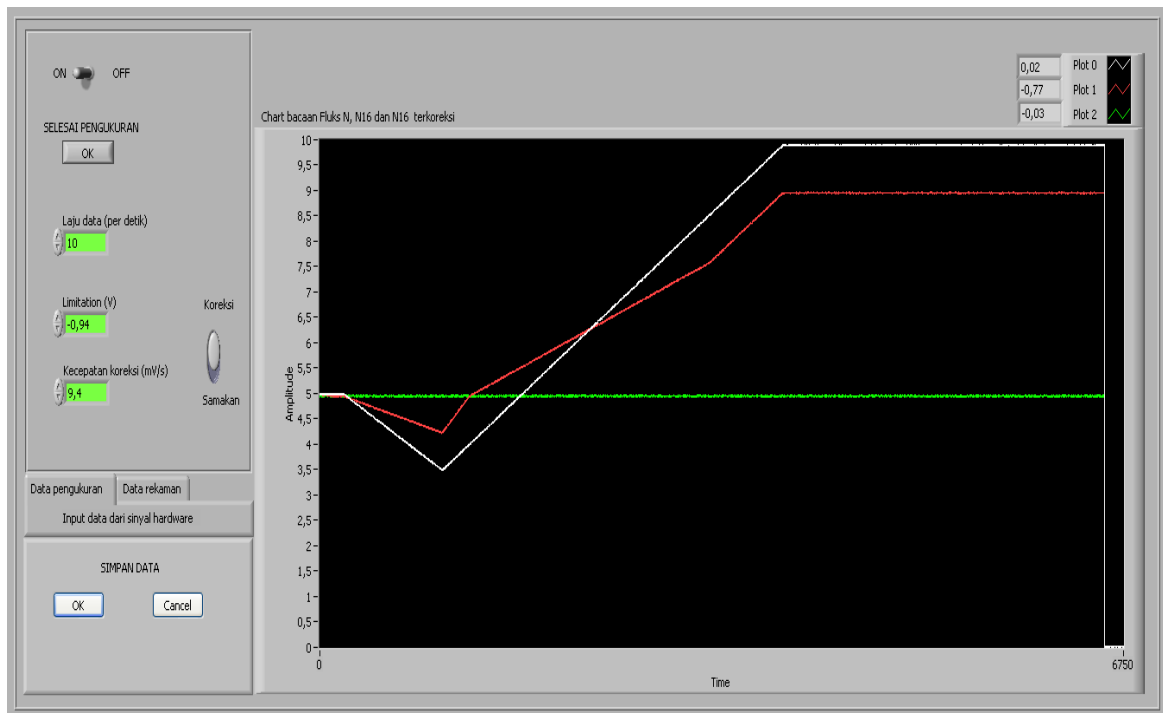


Gambar 4. Diagram Blok Utama





Gambar 5. Case Structure dan Sequent Structure dalam Blok Diagram



Gambar 6. Hasil Pengukuran Sinyal Uji