

PEMBUATAN PROGRAM KOMPUTER POKINIK

Putranto Ilham Yazid

Pusat Penelitian Teknik Nuklir - Badan Tenaga Atom Nasional

ABSTRAK

PEMBUATAN PROGRAM KOMPUTER POKINIK. Telah dibuat program komputer POKINIK yang dapat menghitung reaktivitas batang kendali secara teliti dan andal berdasarkan data percobaan batang kendali jatuh. POKINIK memecahkan persamaan-persamaan kinetika reaktor dan kinetika terbalik dengan metode Runge-Kutta orde 4 yang diprogram secara adaptif. Selain reaktivitas batang kendali, keluaran POKINIK adalah jumlah simpangan dan kuadrat simpangan antara data percobaan dengan hasil penyelesaian persamaan kinetika reaktor, yang merupakan ukuran dari ketelitian dan keandalan hasil perhitungan program POKINIK.

ABSTRACT

POKINIK COMPUTER CODE FOR REACTIVITY CALCULATION. Pokinik is a PC-based computer code written in FORTRAN, which calculates the total control rod worth from the rod drop experiment data. It solves both inverse kinetics and reactor kinetics equations by the method of fourth order Runge-Kutta with adaptive stepsize control techniques. In addition to the reactivity, the code also provide us with the total deviation and square-deviation of experimental data and solution of reactor kinetics equation, which can be used as a measure of reliability and accuracy of the result.

PENDAHULUAN

Saat ini di Sub Bidang Fisika Reaktor PPTN-Bandung terdapat 2 program komputer mengenai kinetika reaktor, yakni POKIN dan KINIK. POKIN [1], merupakan program komputer yang menyimulasikan keadaan rapat neutron $n(t)$ terhadap perubahan reaktivitas $\rho(t)$. Ia menyelesaikan persamaan kinetika reaktor berdasarkan metode Euler, dengan masukan berupa reaktivitas sebagai fungsi waktu $\rho(t)$ dan parameter-parameter reaktor seperti Λ , β_i dan β . Pengaruh-pengaruh reaktivitas balik akibat perubahan suhu, daya dan sebagainya dapat diperhitungkan di dalam $\rho(t)$, sehingga POKIN dapat digunakan untuk mempelajari kelakuan reaktor, yang diwakilkan oleh besaran $n(t)$, terhadap perubahan reaktivitas. Sebaliknya, program KINIK [2] digunakan untuk menyimulasikan / menghitung reaktivitas $\rho(t)$, apabila rapat neutron $n(t)$ diberikan sebagai masukannya. Dengan memasukkan data rapat neutron yang diperoleh dari percobaan batang kendali jatuh, misalnya, KINIK akan menghitung harga reaktivitas sebagai fungsi waktu, dengan cara menyelesaikan persamaan kinetika terbalik berdasarkan metode Runge-Kutta. Program ini terutama berguna untuk menentukan reaktivitas total batang kendali melalui percobaan batang kendali jatuh.

Bila ditilik lebih seksama, kedua program di atas bisa digabungkan untuk memperoleh

satunya program komputer yang dapat menghitung reaktivitas batang kendali secara teliti dan andal. Data percobaan batang kendali jatuh, yang berupa cacah neutron terhadap waktu, diolah oleh KINIK sehingga menghasilkan reaktivitas terhadap waktu. Hasil reaktivitas rerata dari keluaran tersebut selanjutnya menjadi masukan bagi program POKIN, yang kemudian akan menghasilkan harga rapat neutron terhadap waktu. Bila keluaran ini dibandingkan dengan data percobaan dan dihitung jumlah simpangan maupun kuadrat simpangannya, mereka dapat digunakan sebagai patokan dalam menyempurnakan harga reaktivitas yang telah dihitung oleh KINIK tadi. Dengan melakukan pengulangan pada proses perhitungan program POKIN di atas, di mana dalam setiap tahap pengulangan diberikan masukan $\rho_k = \rho_{k-1} \pm \Delta\rho_{k-1}$, sesuai dengan jumlah simpangan antara data percobaan dengan hasil perhitungan / simulasi POKIN, akan dapat diperoleh harga reaktivitas yang semakin teliti.

Proses perhitungan di ataslah yang menjadi dasar dari program POKINIK dalam menghitung reaktivitas batang kendali melalui percobaan batang kendali jatuh. Perbedaan utama yang terdapat di dalam POKINIK adalah bahwa ia menyelesaikan persamaan kinetika reaktor dan kinetika terbalik dengan metode Runge - Kutta orde 4 yang diprogram secara

adaptif. Dengan demikian perhitungan akan semakin teliti walaupun waktu perhitungan (CPU- time) akan semakin tinggi pula.

TEORI

1. Persamaan kinetika reaktor dan kinetika terbalik

Di dalam reaktor titik, persamaan kinetika reaktor dapat dituliskan sebagai [3-4]:

$$\frac{dn(t)}{dt} = \left[\frac{\rho(t) - 1}{\Lambda} \right] n(t) + \sum_{i=1}^6 \lambda_i C_i(t) \text{ dengan } i = 1, \dots, 6 \quad (2)$$

dengan $i = 1, \dots, 6 \quad (2)$

Dengan memasukkan harga reaktivitas terhadap waktu, $\rho(t)$, pemecahan persamaan (1) dan (2) akan dapat digunakan untuk menyimulasikan rapat neutron pada setiap saat t , $n(t)$. Efek-efek reaktivitas balik akibat peracunan Xenon, perubahan suhu daya dan sebagainya dapat diperhitungkan pula ke dalam besaran $\rho(t)$ tersebut di atas.

Program POKINIK akan menyelesaikan persamaan (1) dan (2) dengan metode adaptif Runge-Kutta orde 4, dengan menganggap reaktivitas berubah secara tiba-tiba dari nol ke harga negatif tertentu dan tetap seperti itu selama simulasi berlangsung. Pengaruh reaktivitas balik sama sekali tidak diperhitungkan. Dengan demikian POKINIK sebenarnya akan menyimulasikan keadaan rapat neutron di dalam suatu percobaan batang kendali jatuh, yang dilaksanakan pada saat reaktor berada dalam keadaan dingin dan bersih.

Di pihak lain, penyelesaian persamaan kinetika-terbalik, yakni :

$$\rho(t) = \left[\frac{dn(t)}{dt} - \sum_{i=1}^6 \lambda_i C_i(t) \right] \frac{\Lambda}{n(t)} + 1 \quad (3)$$

akan memungkinkan dilaksanakannya perhitungan harga reaktivitas terhadap waktu $\rho(t)$ apabila harga rapat neutron $n(t)$ diketahui. Program POKINIK menyelesaikan persamaan (3) juga dengan metode adaptif Runge-Kutta orde 4, dimana harga $\frac{dn(t)}{dt}$ dihitung berdasarkan data yang diperoleh dari percobaan batang kendali jatuh.

2. Adaptif Runge-Kutta orde 4

Suatu persamaan diferensial biasa orde satu yang memiliki bentuk : $y' = f(x,y)$ dapat

diselesaikan dengan metode Runge-Kutta orde 4, sebagai berikut :

$$y_{n+1} = y_n + 1/6 (k_1 + 2K_2 + 2K_3 + K_4) \text{ di mana}$$

$$y_{n+1} = y(x_{n+1}); y_n = y(x_n);$$

$$K_1 = h f(x_n, y_n);$$

$$K_2 = h f(x_n + h/2, y_n + k_1/2)$$

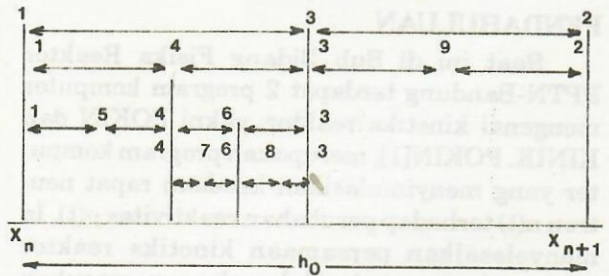
$$K_3 = h f(x_n + h/2, y_n + k_2/2)$$

$$K_4 = h f(x_n + h, y_n + k_3)$$

$$h = x_{n+1} - x_n$$

Dengan demikian $y' = f(x,y)$ dapat diselesaikan dalam selang $a \leq x \leq b$ asalkan harga $y(a)$ diketahui.

Runge-Kutta diyakini merupakan satu metode penyelesaian persamaan diferensial yang memberikan hasil yang amat teliti, walaupun operasi matematik yang dibutuhkannya cukup menyita waktu kerja komputer. Salah satu cara untuk lebih meningkatkan ketelitian hasil penyelesaian dengan metode di atas adalah dengan cara memrogramnya secara adaptif, yang dapat dijelaskan dengan bantuan gambar berikut ini :



Gambar 1 : Proses pembagian sub selang di dalam pemrograman adaptif

Andaikan ingin dicari penyelesaian persamaan diferensial pada $x = x_{n+1}$, yaitu di titik 2 yang berjarak h_0 dari titik 1 dan harga penyelesaian persamaan diferensial tersebut di titik 1 telah diketahui sebelumnya. Mula-mula dicari penyelesaiannya dengan harga $h = h_0$ setelah itu selang 1-2 dibagi dua, menjadi 1-3 dan 3-2. Hitung penyelesaian di titik 2 dengan $h = 1/2h_0$. Kemudian bandingkanlah hasil keduanya, dan apabila beda relatifnya sesuai dengan ketelitian yang diinginkan maka perhitungan langsung berakhir, dengan hasil adalah yang diperoleh dari perhitungan yang menggunakan $h = 1/2h_0$. Sebaliknya, apabila beda relatifnya masih terlalu besar, selang 1-3 kembali dibagi 2, menjadi selang-selang 1-4 & 4-3. Sekarang bandingkanlah hasil penyelesaian di titik 3 dengan menggunakan $h = 1/2h_0$, yakni pada selang 1-3,

dan $h = 1/4h_0$ (selang 1-4 dan 4-3). Bila ketelitian belum juga memuaskan, bagilah selang 1-4 menjadi 1-5 dan 5-4, yaitu $h = 1/8h_0$. Kali ini bandingkanlah penyelesaian di titik 4 dengan menggunakan $h = 1/4h_0$ dan $h = 1/8h_0$. Misalnya saat ini ketelitian sudah memuaskan, berarti telah diperoleh penyelesaian di titik 4 yang merupakan hasil perhitungan dengan memakai $h = 1/8h_0$. Ia kemudian akan digunakan sebagai harga awal dalam perhitungan tahap berikutnya.

Selanjutnya bandingkanlah penyelesaian di titik 3 dengan menggunakan $h = 1/4h_0$ dan $h = 1/8h_0$ (di dalam selang 4-6 dan 6-3). Andaikanlah ketelitian tidak tercapai maka selang 4-6 dibagi 2 menjadi selang-selang 4-7 dan 7-6. Sekarang bandingkanlah penyelesaian di titik 6 dengan memakai $h = 1/8h_0$ dan $h = 1/16h_0$. Anggaplah ketelitian telah dicapai, sehingga kita telah memperoleh harga penyelesaian di titik 6, yakni hasil penyelesaian persamaan diferensial dengan $h = 1/16h_0$. Perhitungan lalu diteruskan dengan membagi dua selang 6-3 menjadi 6-8 & 8-3. Dengan memakai $h = 1/8h_0$ dan $h = 1/16h_0$ bandingkanlah penyelesaian di titik 3. Anggap pula ketelitian telah diperoleh maka saat ini telah dapat diketahui penyelesaian persamaan di titik 3, yakni yang dihitung dengan $h = 1/16h_0$. Selanjutnya bagilah selang 3-2 menjadi 3-9 dan 9-2. Kemudian gunakanlah $h = 1/2h_0$ dan $h = 1/4h_0$ untuk mengetahui ketelitian penyelesaian persamaan di titik 2. Apabila ketelitian belum memuaskan proses perhitungan yang telah diterangkan di atas kembali diulang, untuk selang 3-2, sampai akhirnya penyelesaian di titik 2 diperoleh dengan ketelitian yang tinggi.

Dua keuntungan sekaligus diperoleh dengan cara pemrograman adaptif di atas yakni ketelitian yang tinggi serta jumlah operasi matematik yang sedikit mungkin. Hal terakhir ini dimungkinkan karena sub selang di dalam selang-selang $x_{n-1} - x_n$, $x_n - x_{n+1}$ dan seterusnya bergantung dari bentuk persamaan diferensial yang dipecahkan. Bila harga penyelesaian di titik x_n dan x_{n+1} sangat berbeda jauh, yaitu di dalam daerah kurva penyelesaian yang curam, jumlah sub selang juga akan tinggi. Sebaliknya pada daerah-daerah kurva penyelesaian yang landai tidak diperlukan jumlah sub selang yang banyak untuk memperoleh penyelesaian dengan ketelitian yang sama.

POKINIK dapat membagi satu selang penyelesaian sampai maksimum 256 sub selang. Ini berarti apabila $h = 1/256h_0$ telah tercapai, penyelesaian di titik tersebut akan langsung

dianggap benar, walaupun ketelitian tidak tercapai. Hal ini dimaksudkan agar tidak terjadi iterasi tak berhingga kali.

3. Penghalusan data percobaan

Sebagai akibat dari pemrograman adaptif di atas, dalam menyelesaikan persamaan (1), (2) dan (3) tidak hanya dibutuhkan harga-harga $n(t)$ dan $\frac{dn(t)}{dt}$ di titik-titik batas tiap selang, melainkan juga di titik-titik di antaranya. Pada titik-titik batas, harga $n(t)$ dapat diperoleh dari

data percobaan, demikian pula $\frac{dn(t)}{dt}$. Di

titik-titik di antara suatu batas selang, POKINIK melakukan interpolasi dengan metode *piecewise polynomial*. Ini berarti pula bahwa sebelum data percobaan digunakan sebagai masukan bagi persamaan (3) terlebih dahulu dilakukan penghalusan data. Hal tersebut dilakukan sebagai berikut: Andaikan ingin dicari penyelesaian persamaan (3) di titik t_{n+1} . Pertama dicari polinomial terbaik dengan cara kuadrat-terkecil berdasarkan 10 data di sekitar t_n yakni $t_{n-1}, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+8}$. Polinomial terbaik adalah yang jumlah kwadrat simpangan antara data percobaan dengan hasil yang dihitung dari persamaan polinomial yang koefisiennya dicari dengan cara kuadrat terkecil adalah minimum. Orde polinomial berkisar antara 0 sampai 9. Bila polinomial terbaik telah diperoleh maka harga $n(t)$ dan $\frac{dn(t)}{dt}$ dihitung berdasarkan polinomial tersebut.

Kriteria polinomial terbaik seperti disebut di atas dimaksudkan agar proses penghalusan data bukan semata-mata untuk memperoleh kurva data yang lebih halus, melainkan juga benar-benar tidak berbeda jauh dengan data aslinya. Pengambilan 10 data di sekitar t pada proses interpolasi di atas dimaksudkan agar keterkaitan antara suatu data dengan data lainnya akan semakin diperhitungkan.

4. Proses iterasi

Seperti telah disinggung di dalam bab I, POKINIK dalam menghitung reaktivitas batang kendali akan membandingkan hasil simulasi, yakni $n(t)$ yang berasal dari hasil pemecahan persamaan (1), dengan data percobaan secara berulang-ulang sampai diperoleh harga reaktivitas yang terbaik, yakni jumlah simpangan antara data percobaan dengan hasil simulasi adalah nol atau beda relatif antara reaktivitas pada tahap iterasi terakhir dengan tahap iterasi sebelumnya adalah minimum. Hal ini

dilakukan dengan cara berikut : Setelah persamaan (3) selesai dipecahkan / dihitung berdasarkan masukan data percobaan, diperoleh harga reaktivitas terhadap waktu, $\rho(t)$. Kemudian dihitung harga reratanya, ρ_0 . Dengan harga tersebut diselesaikanlah persamaan (1), dengan mengambil $\rho(t) = \text{konstan} = \rho_0$. Hasil yang diperoleh, rapat neutron terhadap waktu $n(t)$, lalu dibandingkan dengan data percobaan, dengan cara menghitung jumlah simpangan :

$$S = \sum_{i=1}^N n_i - n(t) \text{ di mana:}$$

N = jumlah data.

n_i = data percobaan ke i ($t = t_i$).

$n(t_i)$ = hasil perhitungan/simulasi pada $t = t_i$.

Kemudian dihitunglah $\Delta \rho_0 = \frac{1}{10} |\rho_0|$. Faktor

pengali 1/10 ini sebenarnya boleh sembarang harga, asalkan tidak terlalu besar atau kecil, sehingga iterasi yang akan dilakukan tidak terlalu kasar ataupun terlalu banyak.

Lalu dihitunglah $\rho_0 = \rho_0 - \Delta \rho_0$, apabila $S < 0$, yakni harga simulasi relatif lebih besar dari data percobaan. Kriteria ini diambil dengan mengingat bahwa pada percobaan batang kendali jatuh, harga ρ_0 adalah negatif. Dengan demikian apabila $S < 0$ berarti bahwa harga simulasi $n(t)$ terlampaui landai dibandingkan dengan data percobaan, yang berarti pula bahwa $|\rho_0|$ masih terlampaui kecil. Tetapi sebaliknya, jika $S > 0$ maka $\rho_1 = \rho_0 + \Delta \rho_0$.

Kemudian simulasi diulang dengan $\rho(t) = \rho_1$ dan hitung kembali harga jumlah simpangan S . Selanjutnya hitung $\rho_2 = \rho_1 - \Delta \rho_0$ jika S tetap negatif atau $\rho_2 = \rho_1 + \Delta \rho_0$ bila S tetap positif. Seluruh tahap iterasi di atas, yang disebut iterasi awal, dilaksanakan sampai harga S berubah tanda dari negatif menjadi positif atau dari positif ke negatif. Dalam tahap iterasi awal tersebut harga $\Delta \rho_0$ tetap tidak berubah karena belum bisa diramalkan sampai kapan harga S berubah tanda. Dengan demikian tidak ada alasan atau tidak berguna mengubah $\Delta \rho_0$ selama proses iterasi awal berlangsung.

Andaikata S telah berubah tanda, mulailah iterasi lanjutan dilaksanakan. Mula-mula dihitung $\Delta \rho_1 = \frac{1}{2} \Delta \rho_0$ dan $\rho_{n+1} = \rho_n + \Delta \rho_1$ bila S berubah tanda dari negatif ke positif (kasus I) dan ρ_{n+1} , jika S berubah tanda dari positif ke negatif (kasus II), di mana ρ_n adalah harga reaktivitas pada akhir tahap iterasi awal. Kemudian simulasi diulang dan hitung

$\Delta \rho_2 = \frac{1}{2} \Delta \rho_1$. Pada kasus I, hitunglah $\rho_{n+2} = \rho_{n+1} + \Delta \rho_2$ jika S bertanda tetap seperti sebelumnya dan $\rho_{n+2} = \rho_{n+1} - \Delta \rho_2$ bila S berubah tanda. Sedangkan pada kasus II, ambil $\rho_{n+2} = \rho_{n+1} - \Delta \rho_2$ jikalau S tak berubah tanda dan $\rho_{n+2} = \rho_{n+1} + \Delta \rho_2$ apabila S berubah tanda.

Demikian proses iterasi lanjutan di atas dilakukan berulang kali, di mana pada setiap tahap dihitung harga $\Delta \rho_k = \frac{1}{2} \Delta \rho_{k-1}$, sampai $S = 0$ atau $|(\rho_k - \rho_{k-1}) / \min(\rho_k, \rho_{k-1})| \leq \epsilon$, ketelitian yang kita inginkan. Harga $\Delta \rho$ pada tahap iterasi ini selalu berubah karena setiap saat harga S dapat berubah tanda, sehingga seakan-akan kita sedang menjepit harga ρ yang terbaik yang dicari tersebut. Tabel I dan Gambar 2 akan memperlihatkan contoh yang dapat lebih memperjelas proses iterasi yang dilakukan oleh POKINIK dalam mencari harga reaktivitas yang terbaik.

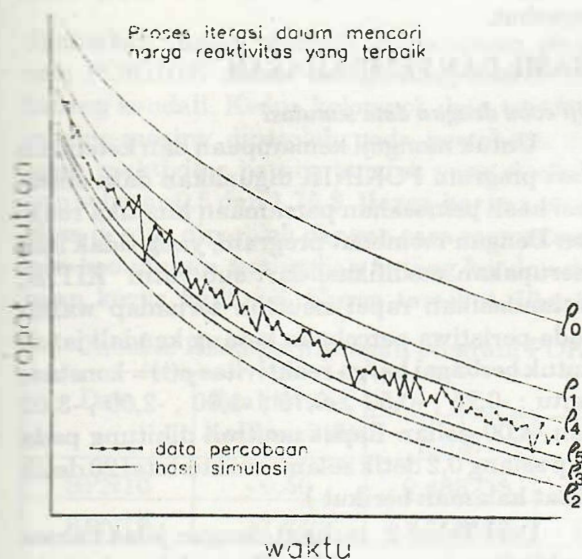
Tabel 1. Contoh harga ρ dan $\Delta \rho$ selama iterasi oleh POKINIK

Tahap iterasi	ρ	$\Delta \rho$	S	Ket.
0	ρ_0	$\Delta \rho_0 = \frac{1}{10} \rho_0$	-	
1	$\rho_1 = \rho_0 - \Delta \rho_0$	$\Delta \rho_0$	-	iterasi awal
2	$\rho_2 = \rho_1 - \Delta \rho_0$	$\Delta \rho_1 = \frac{1}{2} \Delta \rho_0$	+	
3	$\rho_3 = \rho_2 + \Delta \rho_1$	$\Delta \rho_2 = \frac{1}{2} \Delta \rho_1$	+	
4	$\rho_4 = \rho_3 + \Delta \rho_2$	$\Delta \rho_3 = \frac{1}{2} \Delta \rho_2$	-	iterasi lanjutan
5	$\rho_5 = \rho_4 - \Delta \rho_3$	$\Delta \rho_4 = \frac{1}{2} \Delta \rho_3$	+	

Selain menghitung ρ dan S , dalam setiap tahap iterasi juga dihitung jumlah kuadrat simpangan $S_2 = \sum_{i=1}^N (n_i - n(t_i))^2$ yang kelak akan dapat pula digunakan sebagai patokan dalam menentukan harga reaktivitas yang terbaik.

PEMROGRAMAN

Program POKINIK dibuat dengan bahasa FORTRAN-77 dari SINIX. Kompiler yang digunakan adalah FORTRAN-77-CGB-V1.0B buatan Siemens, yang dijalankan di PC-MX2 yang memiliki sistem kerja SINIX, yang merupakan keluarga sistem kerja UNIX. Tidak ada alasan

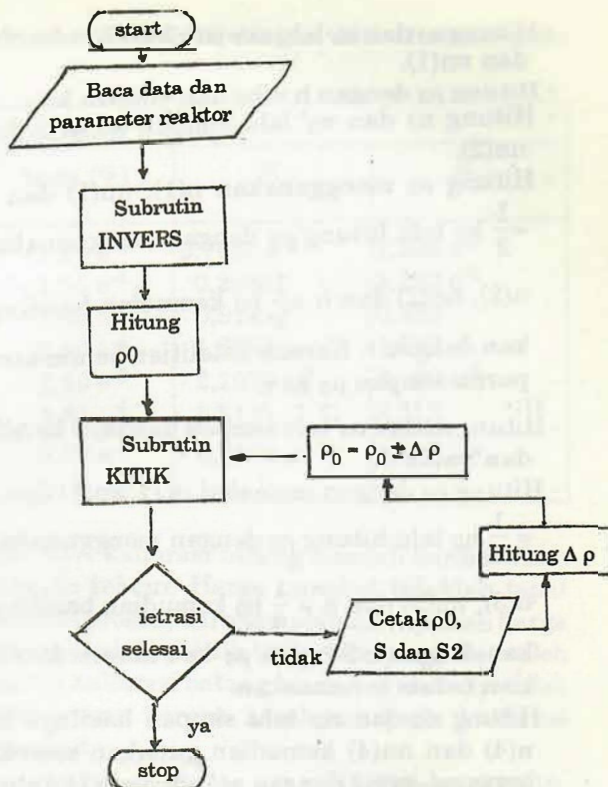


Gambar 2. Contoh proses iterasi dalam mencari harga reaktivitas yang terbaik yang dilakukan oleh POKINIK.

khusus mengapa dipilih bahasa & kompiler tersebut, selain daripada keinginan penulis untuk memanfaatkan kecepatan kerja dari komputer PC-MX2 yang dimiliki oleh Sub Bidang Fisika Reaktor PPTN-Bandung, yang saat ini baru terpasangi kompiler dan bahasa pemrograman tersebut saja.

Secara singkat struktur program POKINIK dapat dijabarkan dalam diagram alir seperti terlihat pada Gambar 3.

Subrutin INVERS menyelesaikan persamaan kinetika terbaliksambil melakukan penghalusan data. Hasil penghalusan data dan reaktivitas awal $\rho(t)$ disimpan ke dalam arsip yang namanya sesuai dengan keinginan pengguna. Subrutin KITIK memecahkan persamaan kinetika reaktor sambil membandingkan hasil simulasinya dengan data percobaan. Setiap kali



Gambar 3. Diagram alir program POKINIK

iterasi diperoleh pula jumlah simpangan S dan jumlah kuadrat simpangan S_2 .

Program POKINIK dibuat seefisien mungkin di mana digunakan variabel bersama (common variable) bagi besaran-besaran yang digunakan secara bersama oleh program utama dan sub-sub rutin. Ini dimaksudkan agar program tidak memakan memori yang terlalu banyak dan tidak pula menyita waktu dalam perpindahan variabel-variabel tersebut ke sub-sub rutin dan sebaliknya.

Algoritma Pemrograman

Efisiensi pemrograman diterapkan pula secara maksimal di dalam pembuatan program adaptif. Pada saat perhitungan berpindah dari sub selang yang lebih kecil, semua besaran yang diperoleh dari perhitungan dengan sub selang yang lebih besar dan masih dapat digunakan pada perhitungan selanjutnya akan disimpan sementara ke dalam variabel deret (array). Besaran-besaran yang disimpan tersebut terutama akan digunakan pada saat perhitungan berpindah ke selang perhitungan yang lebih besar. Sebagai contoh, penyelesaian persamaan (3) pada titik yang berada pada interval 1-2 yang nampak dalam Gambar 1 akan membutuhkan perhitungan dan penyimpanan sebagai berikut :

- Hitung n_1 dan n_1' lalu simpan hasilnya ke $n(1)$ dan $nn(1)$.
- Hitung ρ_2 dengan $h = h_0$ dan simpan ke r .
- Hitung n_3 dan n_3' lalu simpan ke $n(2)$ dan $nn(2)$.
- Hitung ρ_3 menggunakan $n(1)$, $nn(1)$ dan $h = \frac{1}{2} h_0$ lalu hitung ρ_2 dengan menggunakan $n(2)$, $nn(2)$ dan $h = \frac{1}{2} h_0$ kemudian bandingkan dengan r . Karena ketelitian belum sempurna simpan ρ_3 ke r .
- Hitung n_4 dan n_4' lalu simpan hasilnya ke $n(3)$ dan $nn(3)$.
- Hitung ρ_4 dengan memakai $n(1)$, $nn(1)$ dan $h = \frac{1}{4} h_0$ lalu hitung ρ_3 dengan menggunakan $n(3)$, $nn(3)$ dan $h = \frac{1}{4} h_0$ kemudian bandingkan dengan r . Simpan ρ_4 ke r karena ketelitian belum memuaskan.
- Hitung n_5 dan n_5' lalu simpan hasilnya ke $n(4)$ dan $nn(4)$ kemudian gunakan mereka bersama-sama dengan $n(1)$ dan $nn(1)$ untuk menghitung ρ_4 dengan $h = \frac{1}{8} h_0$, yang selanjutnya dibandingkan dengan r . Karena pada saat ini ketelitian telah cukup memadai, gantikan $n(1)$, $nn(1)$ dengan $n(3)$ dan $nn(3)$.
- Hitung ρ_3 dengan menggunakan $n(1)$ dan $nn(1)$ yang baru tersebut dan $h = \frac{1}{4} h_0$ lalu simpan ke r .
- Hitung n_6 dan n_6' lalu simpan hasilnya ke $n(3)$ dan $nn(3)$.
- Hitung ρ_6 dengan menggunakan $n(1)$ dan $nn(1)$ serta $h = \frac{1}{8} h_0$ lalu ρ_3 dengan $n(3)$, $nn(3)$ dan $h = \frac{1}{8} h_0$. Bandingkanlah ρ_3 yang baru tersebut dengan r , dan karena masih kurang teliti simpan ρ_3 tersebut ke r .
- Ulangi proses hitung-simpan di atas sampai diperoleh harga ρ yang terbaik di titik 2.

Algoritma pemrograman di atas tentu akan menyebabkan pembuatan program bertambah rumit. Akan tetapi suatu hal yang pasti adalah bahwa waktu kerja komputer akan semakin singkat, karena tidak ada perhitungan ulang terhadap n dan n' di tiap-tiap titik di dalam suatu selang perhitungan.

Masukan untuk POKINIK

Pokinik membutuhkan masukan berupa data parameter reaktor Λ , β , β_i dan λ_i . Harga

besaran-besaran tersebut sesuai dengan jenis reaktor yang digunakan. Ketelitian perhitungan ditentukan berdasarkan masukan yang berupa selang waktu perhitungan maksimum, h_{\max} yang harganya lebih kecil atau sama dengan selang waktu pencacahan data percobaan. Selang waktu pencacahan data ditentukan oleh masukan-masukan lama percobaan/pengambilan data dan jumlah selang pengambilan data. Demikian pula dengan jumlah data percobaan yang dibutuhkan sebagai masukan.

Pembacaan arsip data masukan dibuat dalam bentuk bebas, dengan menggunakan READ (unit,*), yang akan sangat memudahkan penggunaan dalam pembuatan arsip data tersebut.

HASIL DAN PEMBAHASAN

Uji coba dengan data simulasi

Untuk menguji kemampuan dan ketelitian dari program POKINIK digunakan data simulasi hasil pemecahan persamaan kinetika reaktor. Dengan membuat program, yang tidak lain merupakan modifikasi dari sub rutin KITIK, disimulasikan rapat neutron terhadap waktu pada peristiwa percobaan batang kendali jatuh untuk berbagai harga reaktivitas $\rho(t) = \text{konstan}$, yaitu : -0,25 , -0,50 , -0,75 , -1,00 , -2,00 , -3,00 dan -4,00 dollar. Rapat neutron dihitung pada tiap selang 0,2 detik selama waktu total 20 detik (lihat halaman berikut).

Dari Tabel 2 terlihat dengan jelas bahwa reaktivitas ρ_0 yang dihasilkan oleh sub rutin INVERS praktis tidak berbeda dengan harga reaktivitas yang sebenarnya. Hal tersebut menunjukkan bahwa sub rutin tersebut memiliki kemampuan dan ketelitian tinggi dalam menghitung reaktivitas pada peristiwa percobaan batang kendali jatuh. Proses penghalusan data dan proses pemrograman adaptif yang dilakukannya benar-benar dapat diandalkan.

Di pihak lain reaktivitas yang dihitung secara iteratif oleh sub rutin KITIK juga mampu memperbaiki hasil yang diperoleh dari sub rutin INVERS. Ini dapat dilihat baik dari %-beda maupun dari harga simpangan S dan kuadrat simpangan S_2 . Kedua besaran terakhir tersebut memperlihatkan secara kuantitatif bagaimana ketelitian hasil simulasi sub rutin KITIK dengan data simulasi dalam uji coba tersebut.

Penentuan reaktivitas berdasarkan data percobaan :

Dua kelompok data percobaan yang berasal dari percobaan batang kendali jatuh yang dilakukan di reaktor G.A. Siwabessy Serpong,

Tabel 2. Perbandingan antara hasil perhitungan program POKINIK dengan data simulasi pada berbagai harga reaktivitas.

No	Simulasi (\$)	Pokinik				
		ρ_0 (\$)	ρ di akhir iterasi (\$)	beda (%)	S	S_2
1	- 0,25	- 0,250111431	- 0,250000030	- 1,20 e ⁻⁵	9,7660 e ⁻⁴	1,286 e ⁻⁵
2	- 0,50	- 0,500095367	- 0,500000775	- 1,55 e ⁻⁴	0,2940	2,222 e ⁻²
3	- 0,75	- 0,750094891	- 0,749998798	1,60 e ⁻⁴	- 0,9141	0,482
4	- 1,00	- 0,100001550	- 0,999997199	2,80 e ⁻⁴	1,6530	1,513
5	- 2,00	- 1,999073030	- 2,000000480	- 2,40 e ⁻⁵	- 2,1970 e ⁻³	3,160 e ⁻³
6	- 3,00	- 2,997445580	- 2,999988960	- 3,68 e ⁻⁴	- 0,2141	0,110
7	- 4,00	- 3,995461940	- 3,999998000	- 5,00 e ⁻⁵	- 6,2560 e ⁻²	3,833 e ⁻⁴

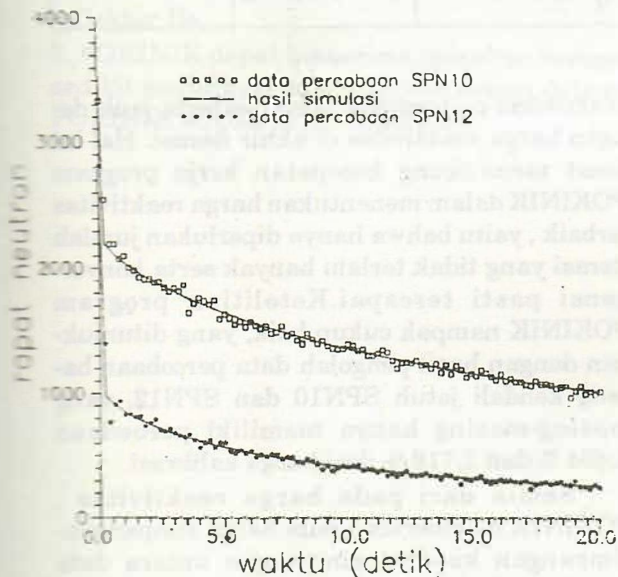
digunakan untuk menguji kemampuan program POKINIK dalam menghitung reaktivitas batang kendali. Kedua kelompok data tersebut masing-masing diperoleh pada percobaan dengan reaktivitas batang kendali yang dijatuhkan sebesar 0,5 dan 1,75 \$. Harga-harga reaktivitas di atas diperoleh dengan cara menyetarakan kedudukan /ketinggian batang kendali dengan kurva kalibrasi. Kurva tersebut dihasil-

kan dari kalibrasi batang kendali berdasarkan metode Inhour. Harga tersebut tidaklah tepat seratus prosen, karena sudah merupakan harga interpolasi dari titik-titik data yang diperoleh dalam kalibrasi batang kendali. Tidak diperoleh keterangan berapa kesalahan pada kalibrasi tersebut.

Data percobaan, yakni SPN10 dan SPN12, masing-masing dengan reaktivitas 0,5 dan 1,75

Tabel 3. Hasil perhitungan program POKINIK terhadap data percobaan SPN10 dan SPN12

Data	Reaktivitas Kalibrasi (\$)	POKINIK				
		ρ_0 (\$)	ρ akhir	Beda (%)	S	S_2
SPN10	- 0,50	- 0,483758	- 0,478499	- 4,494	0,09	227080
SPN12	- 1,75	- 1,758553	- 1,720439	- 1,718	-0,17	246157



Gambar 4. Data percobaan batang kendali jatuh dan hasil perhitungan program POKINIK

Tabel 4. Harga-harga parameter RSG GAS yang dipakai di dalam perhitungan POKINIK dalam mengolah data SPN10 dan SPN12

$$\Lambda = 61,3 e^{-6} \text{ detik. [5]}$$

Kelompok	β_i	λ_i [detik ⁻¹]
1	1,8459 e ⁻⁴	3,872
2	9,0874 e ⁻⁴	1,397
3	2,88936 e ⁻³	3,108 e ⁻¹
4	1,33516 e ⁻³	1,155 e ⁻¹
5	1,51237 e ⁻³	3,174 e ⁻²
6	2,6978 e ⁻⁴	1,272 e ⁻²

\$ serta hasil perhitungan program POKINIK dapat dilihat pada Tabel 3 dan Gambar 4. Tiap titik data diambil dalam selang waktu 0,2 detik.

Hasil tersebut berasal dari pengolahan 101 titik data percobaan. Reaktivitas akhir merupakan harga reaktivitas yang dihitung oleh POKINIK pada saat perbedaan relatif antara harga reaktivitas pada tahap akhir dengan ta-

hap yang sebelumnya lebih kecil dari $1.0e^{-5}$ Harga parameter reaktor yang dipakai dalam perhitungan ini disajikan dalam Tabel 4.

Perbedaan antara reaktivitas kalibrasi dengan hasil perhitungan POKINIK disebabkan oleh beberapa hal, antara lain :

- adanya derau acak selama percobaan berlangsung, yang menyebabkan tidak halusnya data percobaan yang diperoleh.
- ketidaktepatan harga reaktivitas kalibrasi.
- adanya perbedaan konfigurasi reaktor pada

kan harga reaktivitas yang terbaik. Dengan cara mengamati bagaimana bentuk kurva data percobaan serta hasil simulasi POKINIK dan harga-harga S serta S_2 dapatlah secara kualitatif ditentukan harga reaktivitas yang terbaik tersebut.

KESIMPULAN

Metode penghalusan data serta pemrograman adaptif yang diterapkan dalam program POKINIK ternyata sangat andal karena harga

Tabel 5. Harga-harga reaktivitas, simpangan dan kwadrat simpangan dalam tiap tahap iterasi pada pengolahan data percobaan SPN10 dan SPN12.

n	SPN10			SPN12		
	ρ (\$)	S	S_2	ρ (\$)	S	S_2
0	-0,483758	919,67	234645	-1,758553	866,16	258616
1	-0,435382	-8001,31	877510	-1,582698	-3420,60	353567
2	-0,459570	-3408,49	347039	-1,670626	-1181,95	255074
3	-0,471664	-1212,81	243038	-1,714589	-135,57	245662
4	-0,477711	-138,73	227400	-1,736521	370,67	249517
5	-0,480734	392,44	228223	-1,725580	118,87	246917
6	-0,479223	127,44	227115	-1,720085	-8,02	246117
7	-0,478467	-5,66	227072	-1,722833	55,62	246476
8	-0,478845	60,91	227044	-1,721459	23,78	246283
9	-0,478656	27,53	227052	-1,720772	7,92	246198
10	-0,478561	10,92	227065	-1,720428	-0,02	246156
11	-0,478514	2,79	227074	-1,720600	3,93	246178
12	-0,478490	-1,41	227068	-1,720514	1,92	246167
13	-0,478502	0,65	227069	-1,720471	0,94	246161
14	-0,578496	-0,39	227072	-1,720450	0,44	246157
15	-0,478499	0,09	227080	-1,720439	0,17	246157

saat kalibrasi dilakukan dengan pada saat percobaan dilaksanakan.

- adanya perbedaan harga parameter reaktor yang dipakai dalam kalibrasi dengan yang digunakan di dalam POKINIK .

Suatu hal yang menarik adalah apabila diperhatikan harga S dan S_2 dalam tiap tahap iterasi yang dilakukan oleh POKINIK. Tabel 5 menunjukkan bagaimana fluktuasi S dan S_2 selama proses pengulangan berlangsung.

Dari tabel di atas terlihat bahwa walaupun harga reaktivitas ρ seakan-akan tidak jauh berbeda dari satu tahap ke tahap iterasi lain, akan tetapi harga simpangan S dan kuadrat simpangan semakin membaik , yaitu semakin kecil Nampak pula dalam tabel tersebut bahwa harga S yang kecil bukan merupakan jaminan bahwa S_2 akan semakin kecil pula . Hal ini tentunya akan menimbulkan kesulitan dalam menentu-

reaktivitas ρ_0 ternyata tidak berbeda jauh dengan harga reaktivitas di akhir iterasi. Hal ini amat mendukung kecepatan kerja program POKINIK dalam menentukan harga reaktivitas terbaik , yaitu bahwa hanya diperlukan jumlah iterasi yang tidak terlalu banyak serta konvergensi pasti tercapai. Ketelitian program POKINIK nampak cukup baik, yang ditunjukkan dengan hasil pengolahan data percobaan batang kendali jatuh SPN10 dan SPN12, yang masing-masing hanya memiliki perbedaan 4,494 % dan 1,718 % dari harga kalibrasi.

Selain dari pada harga reaktivitas , POKINIK memberikan pula harga simpangan-simpangan kuadrat simpangan antara data percobaan dengan hasil simulasi. Besaran tersebut akan dapat meyakinkan pengguna tentang bagaimana ketelitian hasil perhitungan /penentuan reaktivitas dalam percobaan tersebut.

DAFTAR PUSTAKA

1. Singh, O.P., Mardha, Amil, Kusnowo, Arlinah, "POKIN, a Point Kinetics Computer Codes with Feedback through Dynamic Power Coefficient of Reactivity", IAEA-PPTN/INS/04/018/03/3, 1987.
2. Karkar, K.J., Nabbi, R. and Bormann, H.J., Computer codes for the operational control of the research reactors, ZFK-Jtl- 2037, Jtlch (1986)
3. Ash, Milton S., "Nuclear Reactor Kinetics", McGraw-Hill Inc. (1979)
4. Keepin, G. R., "Physics of Nuclear Kinetics", Addison-Wesley Publ. Comp. Inc., 1965.
5. BATAN, "MPR-30, Safety analysis report", vol 1., September, 1986.

DISKUSI

Henky P.:

Dari tabel yang menyebutkan mengenai ρ_0 dan ρ akhir, disitu anda menyatakan sama, padahal pada baris ke 3 $\rho_0 = 0,75$ dan ρ akhir = 7,5, apakah ini sama ?

Komentar :

- Tulisan reactivity jika di Indonesiakan adalah reaktivitas bukan reaktifitas.
- Kalau presentasi lain kali jangan menutupi transparan, berdiri harus melihat tempat mana yang tidak menutupi layar.

Iham Y. :

Terimakasih atas koreksi dan kritiknya.

Syarip :

1. Apakah hasil (perbedaan hasil eksperimen) di RSG dengan hasil program POKINIK bukan disebabkan oleh pengabaian dari kelompok- kelompok neutron kasip ? (Mestinya >6 kelompok untuk reaktor dengan moderator Be atau ada Be).
2. Apakah input reaktivitas positif dengan fungsi waktu (reaktivitas ramp) bisa diterima oleh POKINIK?

Iham Y.:

1. Bisa saja hal itu benar. Akan tetapi nampaknya tak begitu besar pengaruhnya, karena disain RSG tetap menganggap bahwa neutron kasip hanya 6 kelompok saja, tanpa neutron kasip dari reflektor Be.
2. POKINIK dapat menerima masukan berupa $n(t)$ pada peristiwa ramp reactivity tetapi dengan sedikit modifikasi dalam proses iterasi dalam mencari keaktifan terbaik. Mohon diingat bahwa POKINIK adalah $n(t)$, bukan $\rho(t)$.

DAFTAR PUSTAKA

1. Singh, O.P., Mardha, Amil, Kusnowo, Arlinah, "POKIN, a Point Kinetics Computer Codes with Feedback through Dynamic Power Coefficient of Reactivity", IAEA-PPTN/INS/04/018/03/3, 1987.
2. Kalker, K.J., Nabbi, R. and Bormann, H.,J., Computer codes for the operational control of the research reactors, ZFK-Jtl- 2037, Jtlich (1986)
3. Ash, Milton S., "Nuclear Reactor Kinetics", McGraw-Hill Inc. (1979)
4. Keepin, G. R., "Physics of Nuclear Kinetics", Addison-Wesley Publ. Comp. Inc., 1965.
5. BATAN, "MPR-30, Safety analysis report", vol 1., September, 1986.

DISKUSI

Henky P.:

Dari tabel yang menyebutkan mengenai ρ_0 dan ρ akhir, disitu anda menyatakan sama, padahal pada baris ke 3 $\rho_0 = 0,75$ dan ρ akhir = $7,5$, apakah ini sama ?

Komentar :

- Tulisan reactivity jika di Indonesiakan adalah reaktivitas bukan reaktifitas.
- Kalau presentasi lain kali jangan menutupi transparan, berdiri harus melihat tempat mana yang tidak menutupi layar.

Ilham Y. :

Terimakasih atas koreksi dan kritiknya.

Syarip :

1. Apakah hasil (perbedaan hasil eksperimen) di RSG dengan hasil program POKINIK bukan disebabkan oleh pengabaian dari kelompok- kelompok neutron kasip ? (Mestinya >6 kelompok untuk reaktor dengan moderator Be atau ada Be).
2. Apakah input reaktivitas positif dengan fungsi waktu (reaktivitas ramp) bisa diterima oleh POKINIK?

Ilham Y.:

1. Bisa saja hal itu benar. Akan tetapi nampaknya tak begitu besar pengaruhnya, karena disain RSG tetap menganggap bahwa neutron kasip hanya 6 kelompok saja, tanpa neutron kasip dari reflektor Be.
2. POKINIK dapat menerima masukan berupa $n(t)$ pada peristiwa ramp reactivity tetapi dengan sedikit modifikasi dalam proses iterasi dalam mencari keaktifan terbaik. Mohon diingat bahwa POKINIK adalah $n(t)$, bukan $\rho(t)$.