

---

# Kajian Kombinasi Python Pada Simulator Neuron Untuk Penelitian di Bidang Komputasi Neurosains

Aprilia Dewi Ardiyanti<sup>1\*</sup>, Anis Yuniati<sup>1</sup>

<sup>1,2</sup>Program Studi Fisika, Universitas Islam Negeri Sunan Kalijaga, Jl Marsda Adisucipto  
519739, Indonesia

E-mail: [apriadi379@gmail.com](mailto:apriadi379@gmail.com)

## INTISARI

Simulator NEURON merupakan salah satu simulator yang digunakan dalam pemodelan sel maupun jaringan saraf dalam skala masif. Simulator NEURON dalam perkembangannya dapat ditulis dengan bahasa pemrograman python, baik berdiri sendiri atau dikombinasikan dengan bahasa awal NEURON yaitu *hoc* (high order calculator) dan NMODL. Penulisan python dalam NEURON memberikan akses lebih pada NEURON agar lebih fleksibel dalam penggunaannya. Selain itu, dengan masuknya python dalam simulator NEURON menunjukkan bahwa simulator ini berkembang sesuai dengan bahasa pemrograman yang kelasnya lebih tinggi, lebih mudah dipahami dan lebih banyak digunakan oleh *programmer* pada saat ini.

**Kata kunci:** Komputasi Neurosains, Python, Simulator NEURON

## Abstract

The NEURON simulator is one of the simulators used in modeling cells and neural networks on a massive scale. NEURON Simulator in its development can be written with the Python programming language, either stand alone or combined with the initial NEURON language namely *hoc* (high order calculator) and NMODL. Python writing in NEURON gives NEURON more access to make it more flexible in its use. In addition, the inclusion of python in the NEURON simulator shows that this simulator is developing in accordance with a programming language that is higher class, easier to understand and more widely used by programmers at this time.

**Keyword:** Computational Neurosains, NEURON Simulator, Python

---

## Pendahuluan

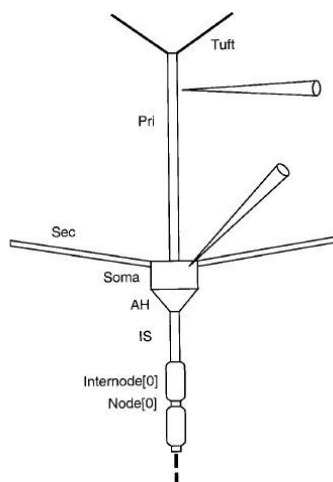
Neurosains merupakan cabang ilmu pengetahuan yang spesifik membahas mengenai sel neuron baik dari segi geometri, struktur maupun transfer informasi yang terjadi antara satu neuron dengan neuron yang lain. Neurosains dalam perkembangan penelitiannya menggunakan dua pendekatan yaitu dengan eksperimen dan permodelan komputasi. Komputasi neurosains berkembang sejak tercetusnya model matematika dari potensial aksi akson cumi-cumi oleh Alan L. Hodgkin dan Andrew F. Huxley. Model neuron tersebut yang dikenal sebagai model Hodgkin-Huxley[1].

Perkembangan komputasi memunculkan software-software untuk mempermudah permodelan, salah satunya adalah simulator NEURON. Simulator NEURON telah dikembangkan sejak 1984 dengan menyediakan pendekatan komputasi yang efisien untuk merekonstruksi, mengatur dan menempatkan model berdasarkan metode numerik yang telah tersedia[2]. Simulator NEURON terintegrasi dengan persamaan-persamaan yang mampu menampilkan mekanisme yang terjadi dalam sel saraf, seperti presinapsis maupun postsinapsis. Kelebihan dari simulator NEURON adalah kecepatan dalam mengeksekusi program yang telah dibuat serta fleksibel untuk pemodelan sel maupun jaringan saraf. Oleh karena itu, simulator ini menjadi simulator yang banyak digunakan dalam bidang komputasi neurosains[3].

Terdapat dua bahasa pemrograman bawaan dari NEURON yaitu *hoc* dan NMODL. Pemrograman dengan *hoc* digunakan untuk membuat model neuron yang terkait dengan morfologi neuron, sifat membran, simulasi maupun perintah untuk menampilkan hasil keluaran data. Simulator NEURON selain menggunakan bahasa pemrograman utama *hoc* juga menggunakan bahasa kedua, yaitu NMODL (*Neuron Model Language*). NMODL adalah bahasa yang digunakan oleh simulator NEURON untuk mendeskripsikan mekanisme reaksi kinetik dalam persamaan differensial maupun aljabar linier. File NMODL yang awalnya mempunyai ekstensi *.mod* akan di *extract* menjadi file dengan ekstensi *.c* dan *.o* agar mampu dikenali oleh simulator NEURON[2].

Bahasa pemrograman *hoc* dan NMODL pada dasarnya telah mampu membuat pemodelan dengan baik, namun diperlukan pengembangan bahasa pemrograman yang lebih fleksibel dan mudah. Bahasa pemrograman python menjadi salah satu alternatif *intrepreter* bagi simulator NEURON. Bahasa pemrograman modern python dipilih dikarenakan beberapa faktor pendukung diantaranya python menyediakan sintaksis yang lebih ekspresif dan fleksibel, mempunyai kapabilitas tinggi dalam debugging, termodulasi, terdapat fasilitas untuk menulis pemrograman yang kompleks, serta mampu terkoneksi dengan GUI lain[4].

Penelitian ini memodelkan dua sel yaitu sel mitral dan sel purkinje. Sel mitral dipilih pada penelitian ini sebagai sel yang dimodelkan dikarenakan data morfologi dan geometri telah berhasil diteliti secara eksperimen. Sel mitral memiliki karakter morfologi tersendiri yaitu, badan sel memiliki satu dendrit primer dan beberapa dendrit sekunder. Dendrit primer panjang dan tidak bercabang. Terdapat dendrit cabang yaitu tuft yang berfungsi untuk menerima masukan sinapsis. Tuft dendrit terdiri dari dua cabang, dan dendrit sekunder juga terdiri dari dua cabang. Selain itu terdapat akson hillock dan akson inisial segmen. Akson tersebut diselubungi oleh lima nodus ranvier dan lima selubung myelin. Morfologi tersebut dapat digambarkan sebagaimana **Gambar 1** dibawah ini[5]



**Gambar 1.** Morfologi Sel Mitral[5]

Sel purkinje merupakan sel yang terletak di bagian *cerrebelum* atau otak kecil, sel ini merupakan salah satu sel yang paling kompleks yang terdapat di otak. Fungsi dari sel purkinje adalah sebagai koordinasi motorik[6]. Sel purkinje memiliki karakteristik dendrit yang sangat banyak yang menyebabkan topologinya berbentuk seperti kipas planar. Dendrit yang terdiri dari kurang lebih 1087 tersebut mengisi hampir seluruh ruang dan terlihat tumpang tindih[7]. **Gambar 2** menggambarkan geometri sel purkinje secara penampang melintang



**Gambar 2.** Morfologi sel purkinje[8]

## Metode Penelitian

Eksperimen menunjukkan bahwa sel mitral maupun sel purkinje telah tervisualisasi dengan menggunakan mikroskop infrared Differential Interference Contras (DIC). Sel mitral mempunyai spesifikasi seperti dalam **Tabel 1** dibawah ini

**Tabel 1.** Bagian-Bagian Sel Mitral

	Soma	Pri	Tuft	Sec	Hillok	Inisial Segmen	Nodus Ranvier	Myelin
Diameter ( $\mu m$ )	20	3.5	0.51	3.4	20:1.5	1.5	1	1.5
Panjang ( $\mu m$ )	25	370	180	500	5	20	1	1000

Sel purkinje mempunyai spesifikasi sebagaimana dalam **Tabel 2** dibawah ini

**Tabel 1.** Bagian-Bagian Sel Purkinje

	Soma	Smoothdend	Spinydend
Diameter ( $\mu m$ )	22	Bervariasi antara 1-6	Bervariasi antara 1, 1.4 dan 2
Panjang ( $\mu m$ )	22	Bervariasi antara 2-44	Bervariasi antara 1-40

Data eksperimen geometri tersebut digunakan untuk menulis pemodelan sel mitral dan sel purkinje. Penulisan pertama dengan menggunakan sintaksis *hoc*. Permodelan yang telah berhasil dimodelkan dengan *hoc* tersebut kemudian dikonversi dalam bentuk lain yaitu bahasa pemrograman python. Penulisan pada python juga menyesuaikan data hasil eksperimen, hanya terdapat perubahan pada sintaksisnya.

Kedua permodelan bisa berjalan dengan baik dan menghasilkan bentuk morfologi maupun geometri sel yang sama, maka dapat diketahui bahwa konversi yang dilakukan dari bahasa pemrogram *hoc* ke python telah dilakukan dengan benar. Tahap selanjutnya adalah menganalisa hasil rekontruksi yang telah dibuat dengan membandingkan perbedaan kedua sintaksis.

### Hasil dan Pembahasan

Sel mitral dengan karakteristik geometri sebagaimana eksperimen berhasil dimodelkan dengan menggunakan dua bahasa pemrograman yaitu bahasa pemrograman *hoc* dan bahasa pemrograman python. Hasil keduanya menunjukkan kesamaan dan kemiripan, sebagaimana terlihat pada **Gambar 3** dibawah ini

a.

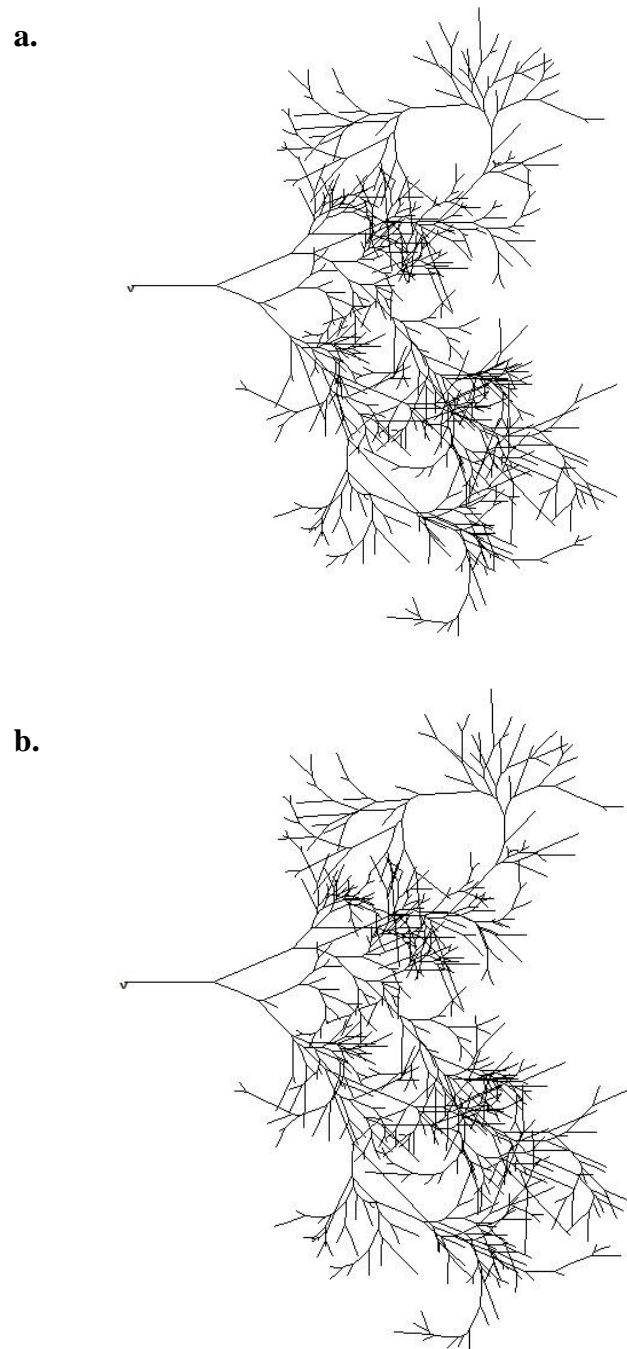


b.



**Gambar 3.** (a) Gambar Pemodelan Sel Mitral dengan Menggunakan bahasa *hoc*, (b) Gambar Pemodelan Sel Mitral dengan Menggunakan bahasa pemrograman Python

Sel purkinje dengan karakteristik geometri sebagaimana eksperimen juga berhasil dimodelkan dengan menggunakan kedua bahasa. Hasil permodelan sebagaimana **Gambar 4** dibawah ini



**Gambar 4.** (a) Gambar Pemodelan Sel Purkinje dengan Menggunakan bahasa *hoc*, (b) Gambar Pemodelan Sel Purkinje dengan Menggunakan bahasa pemrograman Python.

Penulisan kedua pemodelan pada dasarnya hampir sama, hanya terdapat beberapa tambahan dan sintaksis yang berbeda. Simulator NEURON dapat dibuka dengan atau tanpa *Graphical User Interface* (GUI), GUI menunjukkan tampilan grafik yaitu akan ada tampilan jendela main menu NEURON ketika program ini dijalankan. Dalam python untuk memulai NEURON dengan import modul h. Modul h menunjukkan HocObject, sehingga memungkinkan python untuk memasukkan list program model *hoc* didalamnya. Pada *hoc* untuk memulai pemrograman menggunakan perintah

```
load_file("nrngui.hoc")
```

sedangkan untuk memulai program pada python menggunakan perintah sebagai berikut

---

```
from neuron import h, gui
```

Selanjutnya pada setiap awal list program dari python diawali dengan huruf h yang menunjukkan bagian dari objek hoc interpreter.

Pemodelan sel mitral maupun sel purkinje mempunyai tahap awal yaitu membuat bagianbagian sel tersebut seperti soma, akson, nodus ranvier, myelin dan dendrit. Pada *hoc* untuk membuat bagian sel dapat dilakukan dengan singkat yaitu

```
create soma
```

jika bagian sel yang dibuat lebih dari satu maka dapat menuliskan jumlah sel yang hendak dibuat setelahnya

```
create spinydend[1002]
```

sintaksis tersebut menunjukkan bahwa dendrit dengan nama spinydend telah dibuat dengan jumlah 1002. Sedangkan penulisan dalam python dapat dituliskan sebagai berikut

```
soma = h.Section()
```

jika bagian sel yang dibuat lebih dari satu

```
spinydend = [h.Section(name='spinydend[%d]' % i) for i in range(1002)]
```

Tahapan selanjutnya ketika bagian-bagian sel telah dibuat adalah menghubungkan atau membuat tipologi sel. Tipologi sel ini dapat dilakukan dengan menghubungkan antar bagian sel, sehingga terbentuk sel yang sesuai dengan eksperimen dan data. Pada *hoc* untuk membuat tipologi dapat dilakukan dengan menuliskan perintah berikut

```
connect sec dend(0), soma(1)
```

Perintah ini mempunyai makna bahwa soma sebagai induk atau batang sedangkan sec dend sebagai anak atau cabang. Angka 0 dan 1 menunjukkan bagian segmen dari soma dan sec dend yang digabungkan. Pada python untuk membuat tipologi dapat dilakukan dengan menuliskan perintah sebagai berikut

```
sec dend.connect(soma, 1, 0)
```

berbeda dengan *hoc* pada python perintah connect dituliskan ditengah antara anak dan induk.

Geometri dari setiap bagian sel seperti panjang (L), diameter (diam) dan segmen (nseg) dapat dituliskan lebih detail. Pada *hoc* dapat dituliskan dalam perintah

```
Soma {
  nseg=1
  L=22
  diam=22
}
```

Sedangkan pada python dapat ditulis sebagai berikut

```
soma.nseg = 1
soma.L = 22
soma.diam = 22
```

tahapan-tahapan diatas merupakan perbedaan penulisan sintaksis ketika membuat permodelan sel antara menggunakan bahasa pemrograman *hoc* pada NEURON dan bahasa pemrograman python. Perbedaan sintaksis tersebut mengeluarkan hasil pemodelan sel yang sama, sehingga dari penelitian ini dapat diketahui bahwa bahasa pemrograman python dapat dikombinasikan dengan simulator NEURON untuk penelitian neurosains dalam pendekatan komputasi.

## Kesimpulan dan Saran

Pemodelan sel mitral dan sel purkinje berhasil dilakukan dengan menggunakan kombinasi antara simulator NEURON dan bahasa pemrograman Python. Pada pemodelan terdapat beberapa sintaksis yang berbeda antara bahasa *hoc* dan bahasa Python. Hasil Pemodelan menunjukkan geometri dan morfologi yang sama. Hasil pemodelan ini dapat digunakan sebagai rujukan penelitian neurosains yang dalam kajian komputasi, khususnya menggunakan kombinasi simulator NEURON dan Python.

## Daftar Rujukan

- [1] E. De Schutter, "Why Are Computational Neuroscience and Systems Biology So Separate?," *PLOS Computational Biology*, vol. 4, p. e1000078, 2008.
- [2] M. L. Hines and N. T. Carnevale, "The NEURON Simulation Environment," *Neural Computation*, vol. 9, pp. 1179-1209, 1997/08/01 1997.
- [3] M. Migliore, C. Cannia, W. W. Lytton, H. Markram, and M. L. Hines, "Parallel network simulations with NEURON," *Journal of Computational Neuroscience*, vol. 21, p. 119, 2006/05/26 2006.
- [4] T. Oliphant, *Python for Scientific Computing* vol. 9, 2007.
- [5] G. Y. Shen, W. R. Chen, J. Midtgaard, G. M. Shepherd, and M. L. Hines, "Computational Analysis of Action Potential Initiation in Mitral Cell Soma and Dendrites Based on Dual Patch Recordings," *Journal of Neurophysiology*, vol. 82, pp. 3006-3020, 1999/12/01 1999.
- [6] S. Masoli, S. Solinas, and E. D'Angelo, "Action potential processing in a detailed Purkinje cell model reveals a critical role for axonal compartmentalization," *Frontiers in Cellular Neuroscience*, vol. 9, 2015-February-24 2015.
- [7] T. Hirano, "Purkinje Neurons: Development, Morphology, and Function," *The Cerebellum*, vol. 17, pp. 699-700, 2018/12/01 2018.
- [8] S. Gasparini and M. Migliore, "Action Potential Backpropagation," in *Encyclopedia of Computational Neuroscience*, D. Jaeger and R. Jung, Eds., ed New York, NY: Springer New York, 2015, pp. 1-6.