

IMPLEMENTASI PENGACAKAN DAN PENGURUTAN PADA APLIKASI ALICE

Rachmat Selamat

Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI
Jl. Ir. H. Juanda 96 Bandung 40132

rachmatselametskom@gmail.com

ABSTRAK

Alice adalah salah satu pembelajaran pemrograman paling dasar dengan tampilan yang mudah digunakan. Salah satu fitur yang dimiliki Alice adalah membuat variabel dan dapat diisi secara acak. Kebalikan dari acak adalah pengurutan data. Dalam jurnal ini akan dibahas bagaimana mengimplementasikan pengacakan dan pengurutan pada aplikasi Alice yang digambarkan dengan objek 3D.

Kata-kata kunci : Alice, Pengurutan, Acak, Posisi

1. PENDAHULUAN

Alice adalah program yang dapat diunduh secara gratis (freeware) dan sudah didukung oleh Sun (pembuat bahasa Java) untuk mempelajari pemrograman bahasa Java dengan lebih mudah. Kode program yang ada di Alice dapat juga ditampilkan dalam bentuk bahasa Java, sehingga dapat membiasakan pembuat program dalam pembuatan kode bahasa Java.

Dalam perkembangannya, Alice sudah berkembang sampai versi 3, di mana tampilan hasil yang dibuat dapat divisualkan dalam bentuk 3D. Walaupun penggunaannya mudah, tetapi beberapa konsep dasar pemrograman berbasis objek tetap ada dan dapat diterapkan di Alice. Beberapa konsep yang dapat dipelajari adalah konsep kelas, objek, pewarisan, dan pengkapsulan (encapsulation).

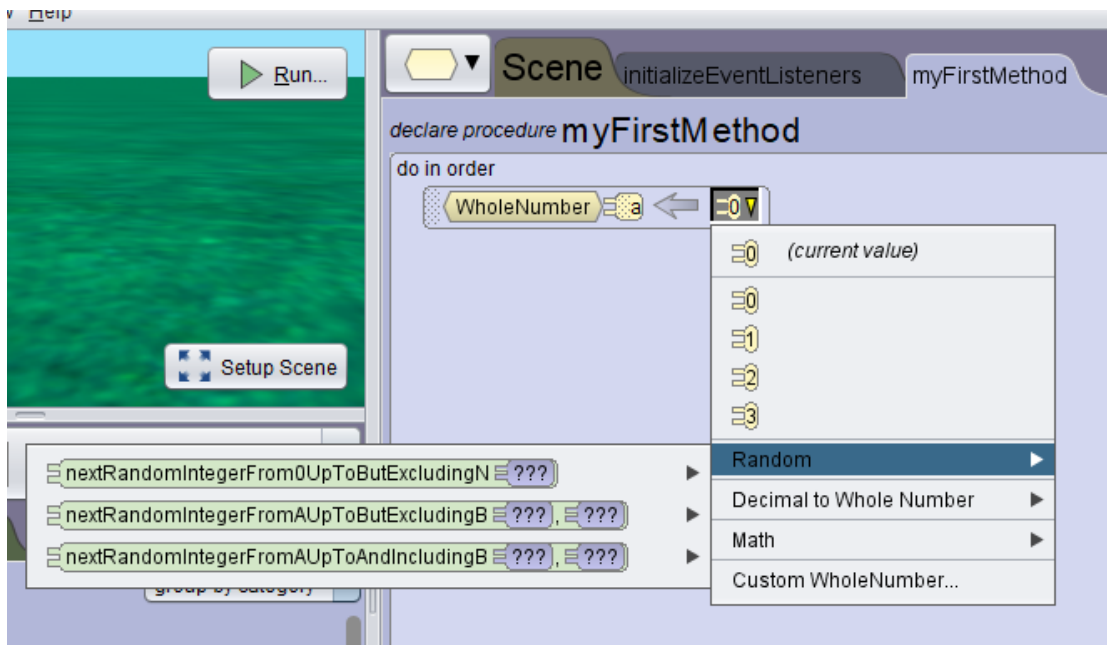
Selain konsep pemrograman berorientasi objek, terdapat juga fitur konsep pemrograman secara umum. Salah satunya tentang pembuatan dan penggunaan variabel (dalam PBO dikenal dengan istilah atribut atau properti). Jenis variabel yang disediakan juga mengenal tipe data umum, seperti tulisan (*text*), bilangan bulat (*WholeNumber*), bilangan desimal (*DecimalNumber*), boolean, *sThing* (sama seperti tipe data object di Java, yang dapat mewakili apapun), serta array. Dalam pemrograman secara umum, array biasa digunakan untuk memetakan posisi antara tampilan di layar dengan proses yang dikerjakan di memori. Untuk bisa memetakan tadi di Alice, tidak seperti di bahasa pemrograman pada umumnya. Pada bab 3 akan dijelaskan bagaimana menerapkannya.

2. FITUR ACAK DI ALICE

Pengumpulan data dalam pemrosesan di program dapat dilakukan dengan beberapa cara, salah satunya adalah dengan membuat data acak. Data acak secara umum dibuat berdasarkan batas tertentu, misalnya dari berapa sampai berapa. Alice juga memiliki fitur ini, Berikut ini beberapa jenis pembuatan angka acak di Alice :

- a. Random untuk bilangan desimal
Dari 0.0 sampai 1.0
Dari ... sampai ...
- b. Random untuk bilangan bulat
Dari 0 excluding sampai z (z tidak masuk)
Dari A excluding sampai z (z tidak masuk)
Dari A including sampai z (z masuk)
- c. Random untuk Boolean :
NextRandomBoolean (antara true atau false)

Cara membuat bilangan acak di alice, pertama buatlah variabel atau perintah yang akan menggunakan bilangan acak tersebut. Untuk angka acak awal, isikan bilangan apa saja kemudian bilangan tersebut diubah menjadi bilangan acak (Random). Contoh mengubah menjadi bilangan acak pada jenis data WholeNumber ada pada Gambar 1.

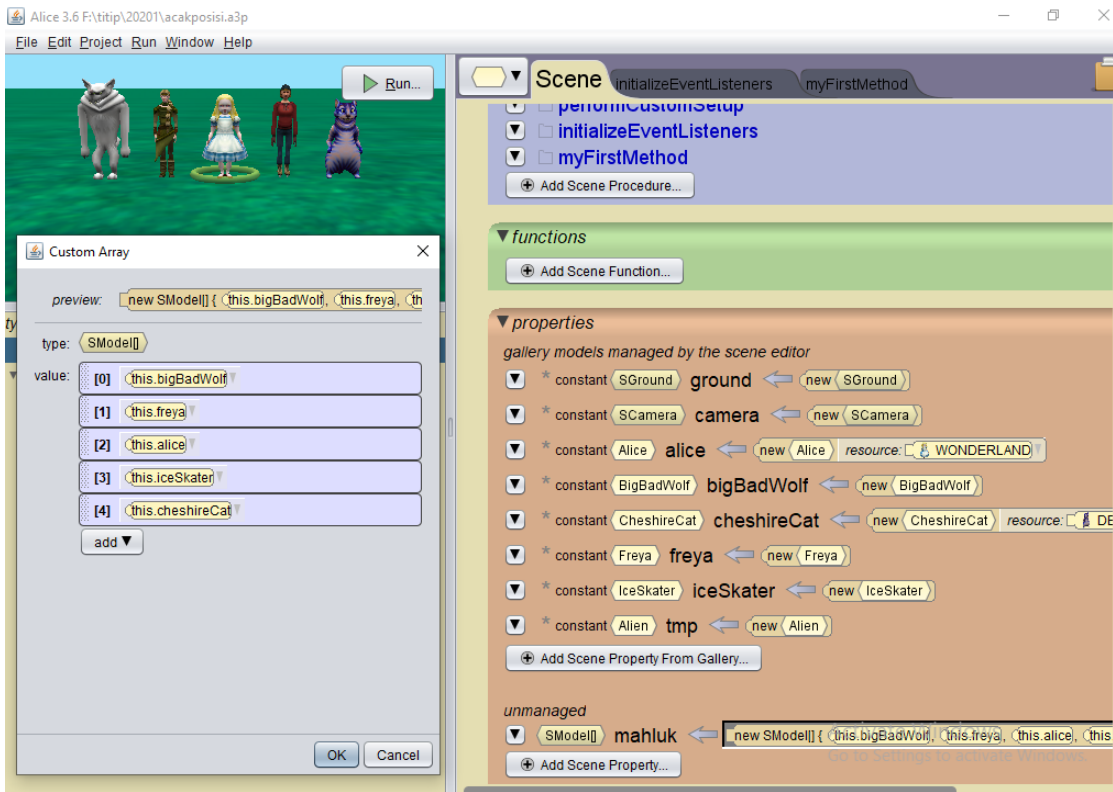


Gambar 1. Contoh Membuat Bilangan Acak

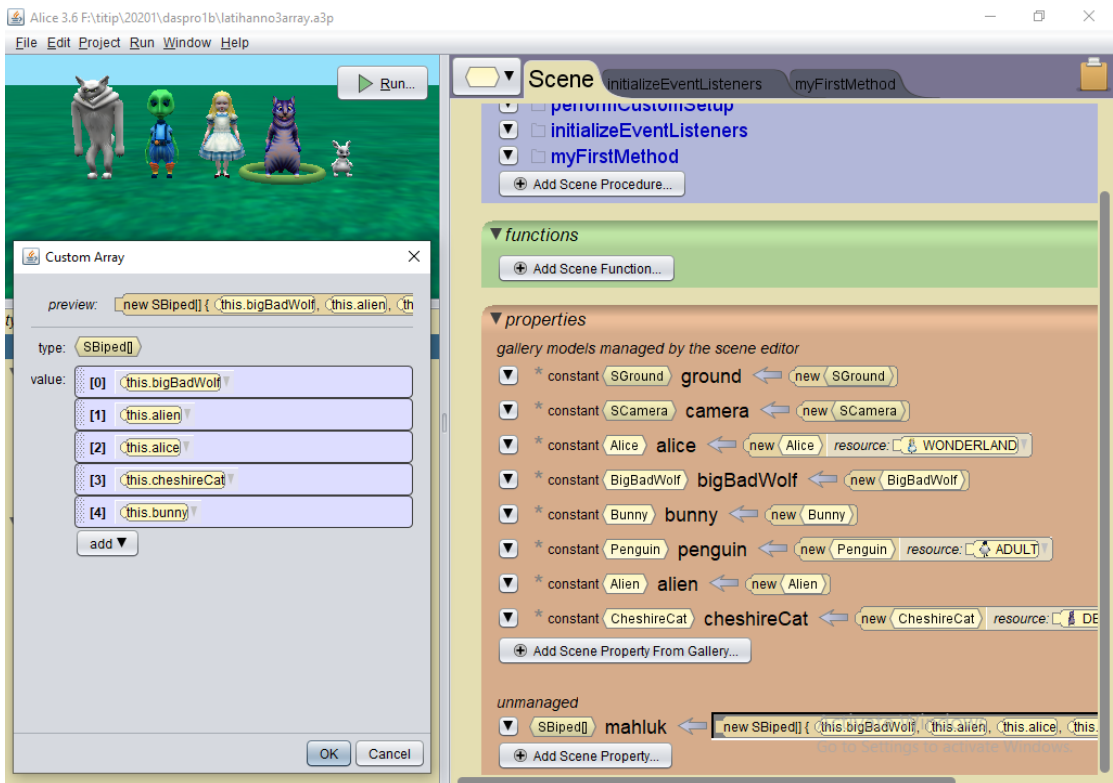
3. INISIALISASI POSISI OBJEK 3D

Salah satu kegunaan array selain untuk menampung data yang banyak, mudah diakses, juga dapat mewakili indexnya sebagai posisi. Kegunaan array terakhir ini yang biasa digunakan untuk memetakan tampilan layar dengan pengaksesan data di memori. Contoh pemakaian, misalnya $data[1]=5$, artinya data ke posisi 2 (dihitung dari 0 indexnya) mewakili data 5. Untuk mencoba menerapkan kegunaan ini di alice, tidak semudah dengan pemrograman biasa.

Untuk menerapkannya, langkah pertama buatlah dulu objek 3d yang berbaris dengan posisi bebas. Kemudian buatlah array secara global (masuk melalui scene->add property) yang menampung objek 3d itu sesuai dengan urutan yang ada di layar awal. Dengan menyesuaikan urutan dengan yang ada di layar, maka index array tersebut sudah mewakili urutan posisi dari objek 3d. Contoh penerapan ini ada pada Gambar 2 dan Gambar 3.



Gambar 2. Contoh Penerapan Inisialisasi Array pada Project1



Gambar 3. Contoh Penerapan Inisialisasi Array pada Project2

4. IMPLEMENTASI PENGACAKAN DAN PENGURUTAN

Pada bagian sebelumnya sudah diberikan contoh inialisasi array untuk 2 project, yaitu project1 dan project2. Project1 akan mengimplementasikan bagaimana cara mengacak data, sedangkan project2 akan mengimplementasikan bagaimana cara mengurutkan data dengan metode minimum sort (selection sort dengan metode minimum).



Gambar 4. Kode untuk Cara Mengacak Data

Dari gambar 4, terlihat bagaimana cara mengacak data. Pertama buatlah counter yang digunakan untuk menyimpan posisi data yang diakses. Kedua buatlah bilangan acak yang menunjukkan posisi data yang ditukar. Periksa apakah posisi keduanya sama atau tidak, apabila tidak maka tukarlah data. Proses ini dikerjakan dari data yang diakses awal (0) sampai jumlah data (dihitung oleh length dari array). Penukaran data dilakukan 2x, yaitu penukaran di array dan di tampilan objek 3d.

```

declare procedure myFirstMethod
do in order
  SBiped ← terpendek ← this.alice
  WholeNumber ← posisipendek ← 0
  WholeNumber ← posisi ← 0
  WholeNumber ← banding ← 0

  count up to this.mahluk.length - 1
  banding ← posisi + 1
  posisipendek ← posisi
  terpendek ← this.mahluk [ posisi ]

  while banding < this.mahluk.length is true
    if this.mahluk [ banding ] getHeight < terpendek.getHeight is true
      terpendek ← this.mahluk [ banding ]
      posisipendek ← banding
    else
      drop statement here
    banding ← banding + 1
  loop

  if this.mahluk [ posisi ] ≠ terpendek is true then
    this.penguin moveTo terpendek add detail
    terpendek moveTo this.mahluk [ posisi ] add detail
    this.mahluk [ posisi ] moveTo this.penguin add detail
    this.mahluk [ posisipendek ] ← this.mahluk [ posisi ]
    this.mahluk [ posisi ] ← terpendek
  else
    drop statement here
  posisi ← posisi + 1
loop
  this.penguin say "selesai" add detail

```

Gambar 5. Kode dalam Pengurutan Data

Dari gambar 5, terlihat bagaimana cara mengurutkan data menggunakan selection sort dengan metode minimum. Cara kerjanya pertama kita membuat variabel counter yang mewakili posisi data yang diakses. Kedua kita mencari posisi data yang objek 3d nya terpendek. Apabila posisi data terpendek tidak sama dengan posisi data yang diakses, maka tukarkan datanya. Sama seperti pengacakan, pengurutan juga menukar datanya 2x yaitu di array dan di tampilan 3d. proses pertama dilanjutkan terus, dari data awal (0) sampai data sebelum yang terakhir.

5. KESIMPULAN

Kesimpulan yang dapat ditarik dari jurnal ini adalah sebagai berikut:

- a. Urutan inisialisasi data objek 3d pada array dapat mewakili posisi awal dari objek 3d
- b. Untuk pertukaran data, harus dilakukan 2x yaitu pada array dan tampilan objek 3d di layar
- c. Pengulangan count di alice tidak menyimpan variabel counter, sehingga harus membuat variabel counter sama seperti while
- d. Sama seperti metode pertukaran data di variabel, pertukaran posisi objek 3d juga membutuhkan objek 3d temporari untuk menampung posisi sementara
- e. Implementasi pengacakan dan pengurutan hanya bisa dikerjakan apabila objek 3d sudah diinisialisasi pada array

6. DAFTAR PUSTAKA

- [1] alice.org.
- [2] Rollings, Andrew; Ernest Adams (2006). *Fundamentals of Game Design*. Prentice Hall.
- [3] Dann, Wanda; Cosgroove, Dennis; Slater, Don; Culyba, Dave; Paoletti, Laura; Tang, Pei (2014). *Alice 3 How To Guide*. Alice Community.