Tugas Kuliah Simulasi dan Pemodelan
Program Studi S3 Teknologi Informatika
Universitas Gunadarma Jakarta

Dosen: A. Benny Mutiara Q.

Analisis Program Simulasi Dinamika Molekuler: Simulasi Brownian Dynamics pada Temperatur Konstan.

Oleh:
Retno Maharesi

Molecular Dynamics (MD)

Problem Molecular Dynamics = Problem mendapatkan solusi gerak partikel yang dipengaruhi oleh persamaan gaya klasik Newton: F= ma. Dalam simulasi, besaran waktu yang seharusnya bersifat kontinu dianggap diskrit, dengan selang waktu konstan h.

MD: solusi dari $md^2 R/dt^2 = -F(R)$

Konservasi energi adalah konsep penting dalam dinamika temperatur dari sistem. Konservasi energi dalam simulasi dapat diperlihatkan, dengan syarat dalam metode numeriknya, persamaan energi potensialnya harus dapat dideferensialkan. Permasalahan diferensiabilitas fungsi energi potensial pada batas kotak latis yang menyebabkan asumsi differensiability tak terpenuhi dapat diatasi dengan menerapkan fungsi potensial Lennard-Jones dan radius *cutoff* r_c pada:

$$Vc(r) = V(r) - V(r_c).$$

<u>Brownian Dynamics</u>

-Merupakan pemodelan simulasi dinamis campuran antara deterministik dan stokastik. Pada Brownian Dynamics beberapa derajat kebebasan hanya direpresentasikan melalui pengaruh stokastik yang dilibatkan secara eksplisit dalam persamaan gerak klasik dari partikel-partikel.
-Studi simulasi dilakukan dalam jerangka kerja *canonical ensemble*. Ide dasar dari simulasi adalah: Pengaruh temperatur konstan pemanasan di suatu ruangan tertutup (tangki), dengan medan gaya stokastik yang bekerja pada partikel-partikel.
-Dalam beberapa metode simulasi Dinamika gerak Brown, system digambarkan dengan persamaan-persamaan Differensial stokastik. Contohnya, persamaan gerak dari suatu partikel yang melakukan gerak Brown adalah:

$$m \frac{dv}{dt} = R(t) - \mu v$$

Persamaan differensial orde 1 di atas, dikenal sebagai persamaan gerak Langevin, yang merupakan persamaan gerak stokastik. Pemanasan tangki dapat direalisasikan melalui gaya stokastik R(t), sehingga menimbulkan sebuah pertanyaan: Apakah sifat yang seharusnya dimiliki R(t), agar nilainya ekwivalen dengan temperatur pemanasan ruang T ?

- Investigasi mengenai R(t) yang akan membawa v ke distribusi klasik invarian Maxwell-Boltzmann, yang diharapkan dalam *canonical ensemble*.
- Mulai dengan kasus 1D.
- Penggunaan teory proses stokastik dan rantai Markov, dapat menunjukkan kalau hal tersebut dapat dicapai jika:

$$P(R) = \frac{1}{\sqrt{(2\pi (R^2))}} e^{-R^2/2R^2}$$

dengan;

- $<R^2> = \mu\, k_B T/h$
- $<R(t)> = 0$
- $<R(t)\, R(0)> = 2\mu\, k_B T\, \delta(t)$

-h : interval waktu untuk integrasi numerik dari persamaan gerak.

Pemilihan R(t) membawa pada hasil:

$$P(v) = \sqrt{\frac{m}{2\mu\, k_B T}}\; e^{\frac{-mv^2}{2k_B T}}$$

Algoritma untuk mensimulasikan gerak Brown Dinamis, secara umum menggunakan algoritma berikut.

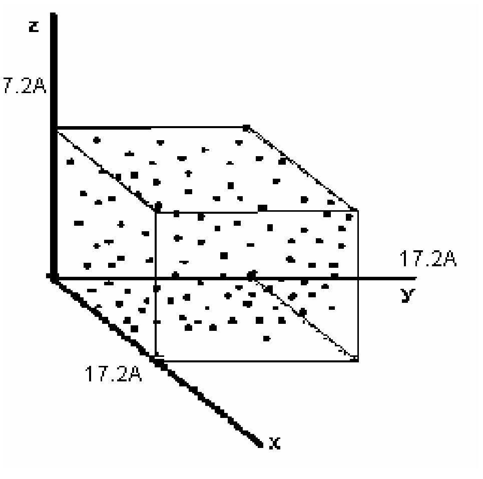
Algoritma :
1. Tetapkan posisi awal dan kecepatan awal dari partikel.
2. Bangkitkan bilangan acak dengan distribusi Gauss untuk rata-rata= 0 dan variansi = R^2 sebagaimana yang dijelaskan di atas. Proses ini akan memberikan R(t).
3. Integralkan persamaan gerak dengan nilai R yang diperoleh di langkah 2 dan tentukan posisi baru berikut kecepatan dari partikel.
4. kembali ke langkah 2.

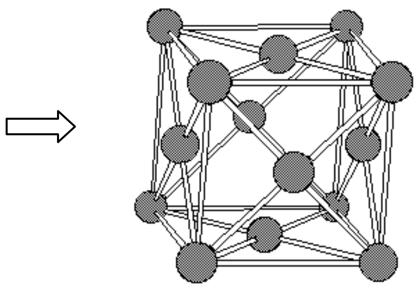
Implementasi langkah 3 pada program simulasi dilakukan dengan menggunakan bentuk penjumlahan dari algoritma Verlet A3 berikut:

Algoritma A3. NVE MD Velocity Form:

(i) Tentukan posisi awal r_i^1
(ii) Tentukan kecepatan awal v_i^1
(iii) Hitung posisi semua parttikel pada saat n+1 dengan formula:
$r_i^{n+1} = r_i^n + hv_i^n + 1/2m^{-1}h^2F_i^n$.
(iv) Hitung kecepatan semua parttikel pada saat n+1 dengan formula:
$v_i^{n+1} = v_i^n + h(F_i^{n+1} + F_i^n)/2m$.

Pada mulanya, partikel diletakkan pada permukaan-pusat dari latis kubus sehingga mengakibatkan jumlah partikel merupakan kelipatan 4→ 256 = 4^4.





latis kubus

Nilai kecepatan dianggap berdistribusi normal. Besaran kecepatan partikel diatur sedemikian rupa secara periodik agar dihasilkan level energi tertentu yang dapat ditoleransi, yaitu setara dengan rata-rata dari temperatur sistem.

Catatan Listing Program Simulasi:

Analisis dari *source code* program simulasi Brownian dynamics dilakukan dengan freeware yang diambil dari internet. Karena alasan tersebut software tidak dapat digunakan untuk mengeksekusi kode secara keseluruhan. Pada tugas simulasi ini kami menggunakan dua freeware: **Force 2.08** (Fortran compiler and editor) dan pemrograman basic untuk Windows, **Liberty. Bas**. Dari force 2.08 diperoleh analisis source code seperti yang terungkap di bawah ini. Oleh karena permasalahan teknis pada Force 2.08, hasil output tidak bisa ditampilkan maka kami menggunakan **Liberty.bas** untuk menjalankan program dengan memperhatikan hasil analisis dengan **Force**.

Algoritma yang digunakan dalam listing program:

(Implementasi Langkah 1)

```
DEFINITION OF THE  SIMULATION PARAMETERS
SET THE ORDER PARAMETER
SET UP FCC LATTICE FOR THE ATOMS INSIDE THE BOX
```

(Implementasi Langkah 2 - 4)

```
ASSIGN BOLTZMANN DISTRIBUTED VELOCITIES AND CLEAR THE FORCES

START OF THE ACTUAL BROWNIAN DYNAMICS PROGRAM:
THE EQUATIONS OF MOTION ARE INTEGRATED USING THE 'SUMMED FORM'
THE STOCHASTIC PART IS FOLLOWS: AT REGULAR TIME INTERVALS THE VELOCITIES ARE
REPLACED BY VELOCITIES DRAWN FROM A BOLTZMANN DISTRIBUTION AT SPECIFIED
TEMPERATURE.

        ADVANCE POSITIONS ONE BASIC TIME STEP
        APPLY PERIODIC BOUNDARY CONDITIONS
        COMPUTE THE PARTIAL VELOCITIES
        COMPUTES THE FORCES ON THE PARTICLES

        COMPUTE THE VELOCITIES
        COMPUTE THE KINETIC ENERGY
        COMPUTE THE AVERAGE VELOCITY
        REPLACE THE VELOCITIES
```

Output Simulasi

```
        COMPUTE VARIOUS QUANTITIES:
        EK     = Energi Kinetik
        EPOT     = Energi Potensial
        ETOT     =Energi total = EK  +  EPOT
        TEMP   =  Temperatur
        PRES  =  Tekanan
        VEL = Kecepatan
        RP   =  (COUNT /  256.0) *100.0
```

Hasil analisis program menggunakan Force 2.08 mengungkapkan beberapa masalah yang terdapat di dalam program, sebagai berikut:

```
C:\WINDOWS\TEMP\cccWYOfb.o: In function `MAIN__':
//F/browniandynamics1.f:86: undefined reference to `ranset_'
```

```
C:\WINDOWS\TEMP\cccWYOfb.o: In function `mxwell_':
//F/browniandynamics1.f:307: undefined reference to `ranf_'
//F/browniandynamics1.f:308: undefined reference to `ranf_'
```

Hal ini berarti, subroutine RANSET(ISEED) pada baris kode call ranset (Iseed) yang hasilnya akan digunakan pada dengan CALL MXWELL (VH,N3,H,TREF) perlu ditambahkan.

Pembangkitan Bilangan Acak dalam FORTRAN 77

Pembangkit bilangan acak FORTRAN 77 yang diberikan oleh para vendor menggunakan CRAY Y-MP UNICOS. Subrutin RANSET dengan argumen integer digunakan untuk mempersiapkan dan mengubah seed; dengan fungsi presisi tunggal yang disebut dengan RANF untuk membangkitkan bilangan acak. RANF tidak memerlukan argumen apapun, berikut ini contoh penulisan kodenya:

```
      INTEGER ISEED
      REAL RANF, U
      ...
      CALL RANSET(ISEED)
      U = RANF()
```

Fungsi pada **subprogram RANF** mengembalikan nilai bilangan acak semu yang terdistribusi secara seragam dalam selang (0,1), dengan membuang nilai yang terakhir. Metode yang digunakan adalah perkalian secara kongruensi. Subrutin subprogram RANGET membuat nilai seed terbaru dari RANF agar dapat digunakan lebih lanjut, dan subrutin RANSET menyimpan nilai seed untuk penggunaan lanjut oleh RANF.

struktur:

SUBROUTINE and FUNCTION subprograms
User Entry Names: RANF, DRANF, RANGET, RANSET

Penggunaan:

Dalam sembarang ekspresi aritmatika,

RANF() diatur ke sebuah nilai yang lebih besar dari nol dan kurang dari satu. RANF bertipe REAL.

CALL RANGET(SEED)
CALL RANSET(SEED)

SEED
(REAL untuk kasus CDC, presisi ganda pada kasus lain). Pada saat keluar dari RANGET, nilai SEED akan diatur ke suatu nilai yang menentukan posisi terbaru dalam barisan bilangan acak. Nilai ini mungkin akan digunakan kemudian sebagai suatu argumen dalam memanggil subrutin RANSET untuk memulai pembangkitan bilangan acak dari titik tersebut.

Berikut ini beberapa referensi dari Intel Fortran Libraries, yang diambil dari internet:

RANSET
Portability Subroutine: Sets the seed for the random number generator.
Module: USE IFPORT
Syntax
CALL RANSET (*seed*)
Seed (Input) REAL(4). The reset value for the seed.
Compatibility
CONSOLE STANDARD GRAPHICS QUICKWIN GRAPHICS WINDOWS DLL LIB
See Also: “RANGET”

RANF
Portability Function: Generates a random number between 0.0 and RAND_MAX.
Module: USE IFPORT
Syntax
result = RANF ( )
Results:
The result type is REAL(4). The result value is a single-precision pseudo-random number between 0.0 and RAND_MAX as defined in the C library, normally 0x7FFF 215–1. The initial seed is set by the following:
CALL SRAND(ISEED) where ISEED is type INTEGER(4).
Compatibility
CONSOLE STANDARD GRAPHICS QUICKWIN GRAPHICS WINDOWS DLL LIB

Pada subrutin Mxwell terdapat u1=ranf() dan u2=ranf(), berdasarkan penjelasan di atas, u1 dan u2 dapat disubstitusi dengan rnd(seed).

Lampiran:

1. **Source Code Fortran untuk Simulasi Brownian Dynamics.**

2. **Source Code Liberty Basic untuk Penggalan Simulasi Brownian Dynamics beserta Hasil Eksekusinya**.

1. Listing Program Brownian Dynamics untuk temperatur konstan:

```
c================================================
c
c         BROWNIAN DYNAMICS
c
c         APPLICATION TO ARGON.THE LENNARD-JONES POTENTIAL IS TRUNCATED
c         AT RCOFF AND NOT SMOOTHLY CONTINUED TO ZERO.INITIALLY THE
c         NPART PARTCLES ARE PLACED ON AN FCC LATTTICE.THE VELOCITIES
c         ARE DRAWN FROM A BOLTZMANN DISTRIBUTION WITH TEMPERATURE TREF.
c
c         INPUT PARAMETERS ARE AS FOLLOWS
c
c         NPART          NUMBER OF PARTICLES (MUST BE A MULTIPLE OF four)
c         SIDE           SIDE LENGTH OF THE CUBICAL BOX IN SIGMA UNITS
c         TREF           REDUCED
c         DEN            REDUCED DENSITY
c         RCOFF          CUTOFF OF THE POTENTIAL IN SIGMA UNITS
c         H              BASIC TIME STEP
c         IREP           REPLACEMENT OF THE VELOCITIES EVERY IREP'TH
c                        TIME STEP
c         TIMEMX         NUMBER OF INTEGRATION STEPS
c         ISEED          SEED FOR THE RANDOM NUMBER GENERATOR
c
c         VERSION 1.0 AS OF AUGUST 1985
c
c         DIETER W.HEERMANN
c
c=====================================================
c
       REAL        X(1:786),VH(1:786),F(1:786)
c
       REAL        FY(1:256),FZ(1:256),Y(1:256),Z(1:256)
c
       REAL        H,HSQ,HSQ2,TREF
c
       REAL        KX,KY,KZ
c

      INTEGER    CLOCK,TIMEMX
c
      EQUIVALENCE (FY,F(257)), (FZ,F(513)),(Y,X(257)),(Z,X(513))
c
c================================================
c
c         DEFINITION OF THE  SIMULATION PARAMETERS
c
c================================================
c
                  NPART = 256
                  SIDE = 6.75284
                  TREF = 0.722
                  DEN = 0.83234
                  RCOFF = 2.5
                  H = 0.064
                  IREP = 10
                  TIMEMX = 3
                  ISEED = 4711
c
c================================================
c
c         SET THE ORDER PARAMETER
c
c================================================
c
          WRITE(6,*) 'BROWNIAN DYNAMICS SIMULATION PROGRAM'
          WRITE(6,*) '--------------------------------------------------'
          WRITE(6,*)
          WRITE(6,*) 'NUMBER OF PARTICLES IS     ',NPART
          WRITE(6,*) 'SIDE LENGTH OF THE  BOX IS             ',SIDE
          WRITE(6,*) 'CUT OFF IS                             ',RCOFF
          WRITE(6,*) 'REDUCED TEMPERATUREE IS           ',TREF
          WRITE(6,*) 'BASIC TIME STEP IS               ',H
          WRITE(6,*)
c
          A           =SIDE/4.0
          SIDEH      =SIDE*0.5
          HSQ        =H*H
          HSQ2       =HSQ*0.5
          NPARTM =NPART-1
          RCOFFS =RCOFF*RCOFF
          TSCALE  =16.0/NPARTM
          VAVER    =1.13*SQRT(TREF/24.0)
c         IOF1        =NPART
          IOF2        =2*NPART
          N3           =3*NPART
c
          CALL RANSET (ISEED)------→subroutine tak terdefinisi
c
c
c=====================================================================
c
c         THIS PART OF THE PROGRAM PREPARES THE INITIAL CONFIGURATION
c
c=====================================================================
c
c         SET UP FCC LATTICE FOR THE ATOMS INSIDE THE BOX
c-----------------------------------------------------------------------------------------------
c         IJK =0
          DO  10  LG=0,1
          DO  10  I=0,3
           DO  10 J=0,3
             DO  10 K=0,3
               IJK = IJK + 1
               X(IJK       )=I*A+LG*A*0.5
               Y(IJK       )=J*A+LG*A*0.5
               Z(IJK       )=K*A
10        CONTINUE
          DO 15 LG=1,2
          DO 15 I=0,3
           DO 15 J=0,3
            DO 15 K=0,3
               IJK =IJK +1
               X(IJK         ) =I*A+(2-LG)*A*0.5
               Y(IJK         )=J*A+(LG-1)*A*0.5
15        CONTINUE
c
c         ASSIGN BOLTZMANN DISTRIBUTED VELOCITIES AND CLEAR THE FORCES
c         ================================================================
c
          CALL MXWELL (VH,N3,H,TREF) ----→ terdapat fungsi yang tak terdefinisi
c
          DO 123 I=1,N3
            F(I)=0.0
123       CONTINUE
c
c=====================================================================
c
c         START OF THE ACTUAL BROWNIAN DYNAMICS PROGRAM.
c
c         THE EQUATIONS OF MOTION ARE INTEGRATED USING THE 'SUMMED FORM'
c         (G.DAHLQUIST AND A. BJOERK, NUMERICAL METHODS, PRENTICE HALL
c         1974). THE STOCHASTIC PART IS FOLLOWS: AT REGULAR TIME
```

```
c           INTERVALS THE VELOCITIES ARE REPLACED BY VELOCITIES DRAWN FROM
c           A BOLTZMANN DISTRIBUTION AT SPECIFIED TEMPERATURE.
c           (H.C. ANDERSEN, J.CHEM. PHYS. 72. 2384 (1980) AND W.C. SWOPE.
c           H.C. ANDERSEN, P.H. BERENS AND K.R. WILSON. J. CHEM . PHYS. 76
c           637 (1982).
c
c           VERSION 1.0 AUGUST 1985
c           DIETER W. HEERMANN
c
c-------------------------------------------------------------------------------------------------
c
            DO        200 CLOCK = 1, TIMEMX
c
c                                === ADVANCE POSITIONS ONE BASIC TIME STEP ===
            DO        210        I=1, N3
                      X(I) = X(I) + VH(I) + F(I)
210         CONTINUE
c
c                                ===APPLY PERIODIC BOUNDARY CONDITIONS===
            DO 215     I=1,N3
                      IF (X(I) .LT. 0)      X(I) = X(I) + SIDE
                      IF (X(I) .GT. SIDE)            X(I) = X(I) - SIDE
215          CONTINUE
c
c                                ===COMPUTE THE PARTIAL VELOCITIES===
c
            DO 220    I=1,N3
                      VH(I) = VH(I) + F(I)
220         CONTINUE
c
c           ================================================================
c           =                                                              =
c           =         THIS PART COMPUTES THE FORCES ON THE PARTICLES       =
c           =                                                              =
c           ================================================================
c
c
                      VIR = 0.0
                      EPOT = 0.0
                      DO 225 I=1,N3
                      F(I) = 0.0
225                   CONTINUE
c
c
                      DO 270 I=1,NPART
                         XI = X(I)
                         YI  = Y(I)
                         ZI = Z(I)
                       DO 270 J = I+1, NPART
                         XX = XI -X(J)
                         YY = YI -Y(J)
                         ZZ = ZI - Z(J)
c
                      IF (XX    .LT.       -SIDEH) XX      = XX      +     SIDE
                      IF (XX    .GT.       SIDEH)  XX      = XX       -     SIDE
c
                      IF (YY    .LT.       -SIDEH) YY      = YY      +     SIDE
                      IF (YY    .GT.        SIDEH) YY      =  YY      -     SIDE
c
                      IF (ZZ    .LT.       -SIDEH) ZZ     =  ZZ     +     SIDE
                      IF (ZZ    .GT.        SIDEH) ZZ      =  ZZ      -     SIDE
                      RD  =  XX* XX + YY * YY + ZZ* ZZ
                      IF  (RD   .GT.   RCOFFS )   GOTO 270
c
                      EPOT = EPOT + RD ** (-6.0) - RD ** (-3.0)
                      R148  = RD ** (-7.0) - 0.5 * RD ** (-4.0)
                      VIR     = VIR -      RD* R148
```

```
                KX      =  XX      * R148
                F(I)     =  F(I)   +   KX
                F(J)     =  F(J)  -  KX
                KY       = YY    * R148
                FY(I)    =  FY(I)   +  KY
                FY(J)    =  FY(J)   -  KY
                KZ       = ZZ         * R148
                FZ(I)    =  FZ(I)   +  KZ
                FZ(J)   =  FZ(J)   -  KZ
270             CONTINUE
           DO   275  I=1, N3
               F(I)    =    F(I)   * HSQ2

275        CONTINUE
c
c       ================================================================
c       =                                                              =
c       =                 END OF THE FORCE CALCULATION                 =
c       ================================================================
c
c                      === COMPUTE THE VELOCITIES===
                DO    300       I=1, N3
                 VH(I)  =  VH(I)  +   F(I)
300        CONTINUE
c
c                      ===COMPUTE THE KINETIC ENERGY ===
                EKIN = 0.0
                DO  305  I= 1, N3
                  EKIN = EKIN + VH(I) * VH(I)
305       CONTINUE
                EKIN = EKIN / HSQ
c
c                      ===COMPUTE THE AVERAGE VELOCITY===
                VEL = 0.0
                COUNT = 0.0
                DO  306  I=1, NPART
                 VX = VH(I)    * VH(I)
                 VY = VH(I + IOF1) * VH (I + IOF1)
                 VZ = VH(I + IOF2) * VH (I + IOF2)
                 SQ = SQRT (VX + VY  + VZ )
                 SQT = SQ / H
                 IF  (SQT  .GT.   VAVER )  COUNT = COUNT +1
                 VEL = VEL +SQ
306       CONTINUE
          VEL = VEL / H
c
          IF ((CLOCK/IREP)* IREP   .EQ.   CLOCK)    THEN
c
c                      ===REPLACE THE VELOCITIES ===
                WRITE(6,*) 'VELOCITY REPLACEMENT' -----------→ tidak terdapat format statement
                CALL MXWELL(VH, N3, H, TREF) ----→ terdapat fungsi yang tak terdefinisi
                EKIN = TREF / TSCALE
          END IF
c
c
c       ===============================================================
c       =                                                             =
c       =                 COMPUTE VARIOUS QUANTITIES                  =
c       =                                                             =
c       ===============================================================
c
        EK     = 24.0  *  EKIN
        EPOT     = 4.0  *  EPOT
        ETOT     = EK  +  EPOT
        TEMP   =  TSCALE * EKIN
        PRES  =  DEN * 16.0 *  (EKIN - VIR) / NPART
        VEL = VEL / NPART
```

```
          RP   =  (COUNT /  256.0) *100.0
          WRITE(6.6000)  CLOCK,EK,EPOT,ETOT,TEMP,PRES,VEL,RP
6000    FORMAT (1I6, 7F15.6)
c
200     CONTINUE
c
c
          STOP
          END
c
c         ====================================================================
c
c         M X W E L L  RETURNS UPON CALLING A MAXWELL DISTRIBUTED DEVIATES
c         FOR THE SPECIFIED TEMPERATURE TREF. ALL DEVIATES ARE SCALED BY
c         A FACTOR
c
c         CALLING PARAMETERS ARE AS FOLLOWS
c         VH        VECTOR OF LENGTH NPART (MUST BE A MULTIPLE OF 2)
c         N3        VECTOR LENGTH
c         H         SCALING FACTOR WITH WHICH ALL ELEMENTS OF VH ARE
c                   MULTIPLIED.
c         TREF      TEMPERATURE
c
c         VERSION  1.0  AS  OF AUGUST  1985
c         DIETER W HEERMANN
c
c         ====================================================================
c
          SUBROUTINE MXWELL(VH,N3,H,TREF)
c
          REAL VH(1:N3)
c
          REAL U1,U2,V1,V2,S,R
c
          NPART  = N3 / 3
          IOF1      = NPART
          IOF2      = 2 * NPART
          TSCALE= 16.0 / (1.0 * NPART - 1.0)
c
          DO     10  I=1,N3,2
c
1           U1   = RANF ()------> tak terdefinisi
            U2   = RANF () -----> tak terdefinisi
c
            V1   =2.0 * U1 - 1.0
            V2   =2.0 * U2 - 1.0
            S     =V1 * V1 +V2 *V2
c
            IF   ( S .GE. 1.0 ) GOTO 1
              R     = -2.0 * ALOG (S) / S
              VH(I)  =  V1 * SQRT (R)
              VH(I+1)= V2 * SQRT (R)
10         CONTINUE
c
          EKIN = 0.0
          SP = 0.0

          DO   20  I=1,NPART
            SP =SP + VH (I)
20        CONTINUE
          SP = SP / NPART
          DO    21 I=1,NPART
              VH (I) = VH (I) - SP
              EKIN  = EKIN + VH (I) * VH (I)
21        CONTINUE
          SP = 0.0
          DO   22  I=IOF1 + 1,IOF2
```

```
           SP = SP + VH (I)
22       CONTINUE
         SP = SP / NPART
         DO  23   I=IOF1 + 1,IOF2
             VH(I) = VH (I) - SP
             EKIN =  EKIN + VH(I) * VH(I)
23     CONTINUE
         SP = 0.0
         DO   24  I=IOF2 + 1,N3
          SP   = SP + VH (I)
24       CONTINUE
         SP = SP / NPART
         DO  25   I=IOF2 + 1,N3
             VH(I) = VH (I) - SP
             EKIN = EKIN + VH (I) * VH (I)
25       CONTINUE
         TS = TSCALE * EKIN
         SC = TREF / TS
         SC = SQRT (SC)
         SC = SC * H
         DO   30  I=1,N3
             VH (I) = VH (I) * SC
30       CONTINUE
         END
```

2. Source codes Liberty Basic

Berikut ini source codes Lyberty Basic untuk program simulasi Brownian Dynamics, setelah membuat beberapa perubahan dari source code aselinya dalam bahasa pemrograman Fortran 77. Kami menjamin source code berikut dapat dijalankan pada software Liberty Basic yang berlisensi. Hal ini dibuktikan dengan mengeksekusi penggalan program:
-Komputasi konstanta di awal program
-Inisial konfigurasi latis dan
-Subrutin Boltzmann-Maxwell.

Karena alasan lisensi software-lah, maka program tidak dapat dikompilasi secara utuh.

-Source codes untuk Brownian Dynamics dalam Liberty Basic secara Utuh

```
DIM VH(1000):DIM F(1000)
  NPART = 256
        SIDE = 6.75284
        TREF = 0.722
        DEN = 0.83234
        RCOFF = 2.5
        H = 0.064
        IREP = 10
        TIMEMX = 3
        ISEED = 4711
print "=================================================="
print"   SET THE ORDER PARAMETER"
print "=================================================="
print "BROWNIAN DYNAMICS SIMULATION PROGRAM"
print " '-------------------------------------------------"
print" NUMBER OF PARTICLES IS "; NPART
print" SIDE LENGTH OF THE  BOX IS   ";SIDE
print" CUT OFF IS  "           ;RCOFF
print" REDUCED TEMPERATUREE IS  ";TREF
print" BASIC TIME STEP IS "; H
    A     =SIDE/4.0
    SIDEH =SIDE*0.5
    HSQ   =H*H
    HSQ2  =HSQ*0.5
    NPARTM=NPART-1
    RCOFFS=RCOFF*RCOFF
    TSCALE=16.0/NPARTM
    VAVER =1.13*(TREF/24.0)^0.5
    IOF1  =NPART
    IOF2  =2*NPART
    N3    =3*NPART

print
"=========================================================
print "  THIS PART OF THE PROGRAM PREPARES THE INITIAL CONFIGURATION"
print
"========================================================="
print "    SET UP FCC LATTICE FOR THE ATOMS INSIDE THE BOX"
print "------------------------------------------------------------------"
```

```
   DIM X(1000):DIM Y(1000): DIM Z(1000)
   IJK =0
    FOR LG= 0 TO 1
    FOR I= 0 TO 3
     FOR J= 0 TO 3
        FOR K= 0 TO 3
           IJK = IJK + 1
           X(IJK      )=I*A+LG*A*0.5
           Y(IJK      )=J*A+LG*A*0.5
           Z(IJK      )=K*A
           PRINT"X(";IJK;")=",X(IJK)
           PRINT"Y(";IJK;")=",Y(IJK)
           PRINT"Z(";IJK;")=",Z(IJK)
              NEXT K
           NEXT J
       NEXT I
NEXT LG

    FOR LG= 1 TO 2
     FOR I= 0 TO 3
     FOR J= 0 TO 3
        FOR K= 0 TO 3
           IJK =IJK +1
           X(IJK        ) =I*A+(2-LG)*A*0.5
           Y(IJK        )=J*A+(LG-1)*A*0.5
           Z(IJK        )= K*A +A*0.5
           PRINT"X(";IJK;")=",X(IJK)
           PRINT"Y(";IJK;")=",Y(IJK)
           PRINT"Z(";IJK;")=",Z(IJK)
                 NEXT K
           NEXT J
       NEXT I
NEXT LG
print"   ASSIGN BOLTZMANN DISTRIBUTED VELOCITIES AND CLEAR THE FORCES"
print"  ======================================================="
  GOSUB[MXWELL]
   FOR I=1 TO N3
       F(I)=0.0
    NEXT I
print"   START OF THE ACTUAL BROWNIAN DYNAMICS PROGRAM."
PRINT"--------------------------------------------------"
FOR T=1 TO TIMEMX
print"   === ADVANCE POSITIONS ONE BASIC TIME STEP ==="
    FOR   I=1 TO N3
        X(I) = X(I) + VH(I) + F(I)
        PRINT"X(";I;")=",X(I)
   NEXT I
    print"    ===APPLY PERIODIC BOUNDARY CONDITIONS==="
    DO until      I=N3
        IF (X(I) < 0) then   X(I) = X(I) + SIDE
        IF (X(I) > SIDE) then   X(I) = X(I) - SIDE
        loop
    print"    ===COMPUTE THE PARTIAL VELOCITIES==="
    FOR I= 1 TO N3
        VH(I) = VH(I) + F(I)
    NEXT I
print"    ======================================================="
```

```
print"   =    THIS PART COMPUTES THE FORCES ON THE PARTICLES    ="
print"   ==================================================="

       VIR = 0.0
       EPOT = 0.0
      FOR  I=1 TO N3
       F(I) = 0.0
      NEXT I
       FOR  I= 1 TO NPART
          XI = X(I)
          YI  = Y(I)
          ZI = Z(I)
     FOR J = 1+I TO NPART
          XX = XI -X(J)
          YY = YI -Y(J)
          ZZ = ZI - Z(J)
       IF (XX    <   -1 *SIDEH) then XX      = XX      +     SIDE
       IF (XX    >    SIDEH)  then XX     = XX        -     SIDE

       IF (YY    <    -1* SIDEH) then YY      = YY      +     SIDE
       IF (YY    >     SIDEH) then YY      =  YY       -     SIDE

       IF (ZZ    <   -1* SIDEH) then ZZ     =  ZZ     +     SIDE
       IF (ZZ    >    SIDEH) then ZZ      =  ZZ       -     SIDE
       RD  =  XX* XX + YY * YY + ZZ* ZZ
       IF  (RD   >   RCOFFS ) then  GOTO 270

       EPOT = EPOT + RD ^(-6) - RD ^ (-3)
       R148  = RD ^ (-7.0) - 0.5 * RD ^(-4.0)
       VIR     = VIR -      RD* R148
       KX      =  XX      * R148
       F(I)      =  F(I)   +   KX
       F(J)     =  F(J)  -  KX
       KY       = YY    * R148
       FY(I)    =  FY(I)   +  KY
       FY(J)    =  FY(J)   -  KY
       KZ        = ZZ        * R148
       FZ(I)    =  FZ(I)   +  KZ
       FZ(J)   =FZ(J) -KZ
270    NEXT J
NEXT I
         FOR  I= 1 TO N3
             F(I)    =    F(I)   * HSQ2
             PRINT"F(";I;")=",F(I)
275       NEXT I

print" ==================================================="
print"        END OF THE FORCE CALCULATION"
print" ==================================================="

    print"     === COMPUTE THE VELOCITIES==="
       for      I=1 to N3
         VH(I)  =  VH(I)  +   F(I)
      next I

    print"     ===COMPUTE THE KINETIC ENERGY ==="
       EKIN = 0.0
       for  I= 1 to N3
          EKIN = EKIN + VH(I) * VH(I)
305     next I
       EKIN = EKIN / HSQ

print" ===COMPUTE THE AVERAGE VELOCITY==="
       VEL = 0.0
       COUNT = 0.0
       for I= 1 to NPART
         VX = VH(I)    * VH(I)
         VY = VH(I + IOF1) * VH (I + IOF1)
         VZ = VH(I + IOF2) * VH (I + IOF2)
         SQ = SQRT (VX + VY  + VZ )
         SQT = SQ / H
         IF  (SQT  >    VAVER ) then COUNT = COUNT +1
         VEL = VEL +SQ
306    next I
     VEL = VEL / H
     IF ((CLOCK/IREP)* IREP   =   CLOCK)    THEN
print"            ===REPLACE THE VELOCITIES ==="
print" VELOCITY REPLACEMENT"
    GOSUB [MXWELL]
       EKIN = TREF / TSCALE
     END IF
print"    ================================================="
print"   =         COMPUTE VARIOUS QUANTITIES        ="
print"  =================================================="
    EK     = 24.0  *  EKIN
    EPOT     = 4.0  *  EPOT
    ETOT     = EK  +  EPOT
    TEMP   =  TSCALE * EKIN
    PRES  =  DEN * 16.0 *  (EKIN - VIR) / NPART
    VEL = VEL / NPART
    RP   =  (COUNT /  256.0) *100.0
print  "CLOCK=",  T;  "ENERGI  KINETIK=",EK;  "ENERGI  POTENSIAL=",  EPOT;  "TOTAL
ENERGI",ETOT
PRINT   "TEMPERATUR=",TEMP;"TEKANAN=",      PRES;"KECEPATAN   PARTIKEL=",   VEL;
"RP=",RP
NEXT T
    STOP
    END

[MXWELL]
     NPART   = N3 / 3
    IOF1    = NPART
    IOF2    = 2 * NPART
    TSCALE  = 16.0 / (1.0 * NPART - 1.0)
    FOR   I=1 TO N3 STEP 2
1     U1   = Rnd(4711)
      U2   = Rnd(4711)
       V1   =2.0 * U1 - 1.0
       V2   =2.0 * U2 - 1.0
       S    =V1 * V1 +V2 *V2
       IF   ( S >= 1.0 )THEN  GOTO 1
          R    = -2.0 * log (S) / S
          VH(I)  =  V1 * R^.5
          VH(I+1)= V2 * R^.5
          10       NEXT I
 EKIN = 0.0
    SP = 0.0
    FOR    I=1 TO NPART
       SP =SP + VH (I)
20    NEXT I
    SP = SP / NPART
    FOR    I=1 TO NPART
          VH (I) = VH (I) - SP
          EKIN  = EKIN + VH (I) * VH (I)
21    NEXT I
    SP = 0.0
   FOR    I=IOF1 + 1 TO IOF2
       SP = SP + VH (I)
22   NEXT I
    SP = SP / NPART
    FOR    I=IOF1 + 1 TO IOF2
         VH(I) = VH (I) - SP
         EKIN =  EKIN + VH(I) * VH(I)
23      NEXT I
    SP = 0.0
    FOR  I=IOF2 + 1 TO N3
     SP   = SP + VH (I)
24    NEXT I
    SP = SP / NPART
    FOR  I=IOF2 + 1 TO N3
         VH(I)    = VH (I) - SP
         EKIN    = EKIN + VH (I) * VH (I)
25    NEXT I
    TS = TSCALE * EKIN
    SC = TREF / TS
    SC = (SC)^0.5
    SC = SC * H
    FOR     I=1 TO N3
         VH (I) = VH (I) * SC
         PRINT"VH(";I;")=",VH(I)
30    NEXT I
RETURN
```

-Source codes untuk Brownian Dynamics-1

```
print "BROWNIAN DYNAMICS"
print "APPLICATION TO ARGON.THE LENNARD-JONES POTENTIAL IS TRUNCATED"
print "AT RCOFF AND NOT SMOOTHLY CONTINUED TO ZERO.INITIALLY THE"
print "NPART PARTCLES ARE PLACED ON AN FCC LATTTICE.THE VELOCITIES"
print "ARE DRAWN FROM A BOLTZMANN DISTRIBUTION WITH TEMPERATURE TREF."
print "INPUT PARAMETERS ARE AS FOLLOWS"
print" NPART  =     NUMBER OF PARTICLES (MUST BE A MULTIPLE OF four)"
print " SIDE  =     SIDE LENGTH OF THE CUBICAL BOX IN SIGMA UNITS"
print "TREF   =     REDUCED"
print " DEN   =     REDUCED DENSITY"
print " RCOFF =       CUTOFF OF THE POTENTIAL IN SIGMA UNITS"
print" H      =  BASIC TIME STEP"
print"  IREP  =     REPLACEMENT OF THE VELOCITIES EVERY IREP'TH"
print"  TIME STEP"
print " TIMEMX =       NUMBER OF INTEGRATION STEPS"
print "ISEED   =     SEED FOR THE RANDOM NUMBER GENERATOR"
print "VERSION 1.0 AS OF AUGUST 1985"
print " DIETER W.HEERMANN"
print "=================================================="

print "=================================================="
print"   DEFINITION OF THE  SIMULATION PARAMETERS"
print "=================================================="
      NPART = 256
      SIDE = 6.75284
      TREF = 0.722
      DEN = 0.83234:PRINT"REDUCED DENSITY=",DEN
      RCOFF = 2.5:PRINT "RCOFF=",RCOFF
      H = 0.064
      IREP = 10:PRINT "REPLACEMENT VELOCITIES EVERY=",IREP;"th-STEP"
      TIMEMX = 3:PRINT"TIME MAX=",TIMEMX
      ISEED = 4711:PRINT"ISEED=",ISEED
print "=================================================="
print"   SET THE ORDER PARAMETER"
print "=================================================="

   print "BROWNIAN DYNAMICS SIMULATION PROGRAM"
   print " '------------------------------------------------"
   print" NUMBER OF PARTICLES IS ", NPART
   print" SIDE LENGTH OF THE  BOX IS   ",SIDE
   print" CUT OFF IS  "          ,RCOFF
   print" REDUCED TEMPERATUREE IS  ",TREF
   print" BASIC TIME STEP IS ", H
   A       =SIDE/4.0
   SIDEH    =SIDE*0.5
   HSQ      =H*H
   HSQ2     =HSQ*0.5
   NPARTM   =NPART-1
   RCOFFS   =RCOFF*RCOFF
   TSCALE   =16.0/NPARTM
   VAVER    =1.13*(TREF/24)^0.5
  IOF1      =NPART
   IOF2     =2*NPART
   N3       =3*NPART
   PRINT"A=",A,"SIDEH =",SIDEH,"HSQ=",HSQ,"HSQ2=",HSQ2
   PRINT"NPARTM=",NPARTM,"RCOFFS=",RCOFFS,"TSCLAE=",TSCALE
   PRINT"VAVER=",VAVER,"IOF1=",IOF1,"IOF2=",IOF2,"N3=",N3
   print "CALL RANSET (ISEED)"
```

-Hasil Eksekusi dari program Brownian Dynamics-1

```
BROWNIAN DYNAMICS
APPLICATION TO ARGON.THE LENNARD-JONES POTENTIAL IS TRUNCATED
AT RCOFF AND NOT SMOOTHLY CONTINUED TO ZERO.INITIALLY THE
NPART PARTCLES ARE PLACED ON AN FCC LATTTICE.THE VELOCITIES
ARE DRAWN FROM A BOLTZMANN DISTRIBUTION WITH TEMPERATURE TREF.
INPUT PARAMETERS ARE AS FOLLOWS
 NPART  =      NUMBER OF PARTICLES (MUST BE A MULTIPLE OF four)
 SIDE  =      SIDE LENGTH OF THE CUBICAL BOX IN SIGMA UNITS
TREF   =     REDUCED
 DEN   =     REDUCED DENSITY
 RCOFF =       CUTOFF OF THE POTENTIAL IN SIGMA UNITS
 H      =  BASIC TIME STEP
  IREP  =     REPLACEMENT OF THE VELOCITIES EVERY IREP'TH
  TIME STEP
 TIMEMX =        NUMBER OF INTEGRATION STEPS
ISEED   =     SEED FOR THE RANDOM NUMBER GENERATOR
VERSION 1.0 AS OF AUGUST 1985
 DIETER W.HEERMANN
===============================================
   DEFINITION OF THE  SIMULATION PARAMETERS
================================================
REDUCED DENSITY=           0.83234
RCOFF=        2.5
REPLACEMENT VELOCITIES EVERY=            10th-STEP
TIME MAX=     3
ISEED=        4711
================================================
   SET THE ORDER PARAMETER
================================================
BROWNIAN DYNAMICS SIMULATION PROGRAM
 '--------------------------------------------------
 NUMBER OF PARTICLES IS     256
 SIDE LENGTH OF THE  BOX IS              6.75284
 CUT OFF IS                  2.5
 REDUCED TEMPERATUREE IS    0.722
 BASIC TIME STEP IS         0.064
A=             1.68821        SIDEH =        3.37642        HSQ=           0.004096        HSQ2=
0.002048
NPARTM=       255          RCOFFS=       6.25          TSCLAE=       0.62745098e-1
VAVER=        0.19599339    IOF1=        256          IOF2=        512          N3=          768
CALL RANSET (ISEED)
```

-Source codes untuk Konfigurasi Awal

```
print "=================================================="
print "  THIS PART OF THE PROGRAM PREPARES THE INITIAL CONFIGURATION"
print =================================================="
print "    SET UP FCC LATTICE FOR THE ATOMS INSIDE THE BOX"
print "------------------------------------------------------------------"
A=           1.68821
DIM X(1000):DIM Y(1000): DIM Z(1000)
   IJK =0
    FOR LG= 0 TO 1
   FOR I= 0 TO 3
    FOR J= 0 TO 3
      FOR K= 0 TO 3
        IJK = IJK + 1
        X(IJK      )=I*A+LG*A*0.5
        Y(IJK      )=J*A+LG*A*0.5
        Z(IJK      )=K*A
        PRINT"X(";IJK;")=",X(IJK)
        PRINT"Y(";IJK;")=",Y(IJK)
        PRINT"Z(";IJK;")=",Z(IJK)
          NEXT K
        NEXT J
     NEXT I
NEXT LG

   FOR LG= 1 TO 2
    FOR I= 0 TO 3
    FOR J= 0 TO 3
      FOR K= 0 TO 3
        IJK =IJK +1
        X(IJK        ) =I*A+(2-LG)*A*0.5
        Y(IJK        )=J*A+(LG-1)*A*0.5
        Z(IJK        )= K*A +A*0.5
        PRINT"X(";IJK;")=",X(IJK)
        PRINT"Y(";IJK;")=",Y(IJK)
        PRINT"Z(";IJK;")=",Z(IJK)
            NEXT K
        NEXT J
     NEXT I
NEXT LG
```

-Hasil Eksekusi dari program untuk Konfigurasi Awal:

```
===========================================================
  THIS PART OF THE PROGRAM PREPARES THE INITIAL CONFIGURATION
===========================================================
   SET UP FCC LATTICE FOR THE ATOMS INSIDE THE BOX
-----------------------------------------------------------------
X(1)=       0
Y(1)=       0
Z(1)=       0
X(2)=       0
Y(2)=       0
Z(2)=       1.68821
X(3)=       0
Y(3)=       0
Z(3)=       3.37642
X(4)=       0
Y(4)=       0
Z(4)=       5.06463
X(5)=       0
Y(5)=       1.68821
Z(5)=       0
X(6)=       0
Y(6)=       1.68821
Z(6)=       1.68821
……………………….
Y(252)=      4.220525
```

```
Z(252)=     5.908735
X(253)=     5.06463
Y(253)=     5.908735
Z(253)=     0.844105
X(254)=     5.06463
Y(254)=     5.908735
Z(254)=     2.532315
X(255)=     5.06463
Y(255)=     5.908735
Z(255)=     4.220525
X(256)=     5.06463
Y(256)=     5.908735
Z(256)=     5.908735
```

-Source codes untuk Maxwell-Boltzmann-1

```
DIM VH(1000)
H=    6.75284
TREF= 0.722
H=  0.064
N3=         768
print"CALL RANSET (ISEED)"
print"    ASSIGN BOLTZMANN DISTRIBUTED VELOCITIES AND CLEAR THE FORCES"
print"    ====================================================="
print"    CALL MXWELL (VH,N3,H,TREF)"
PRINT" SUBROUTINE MXWELL(VH,N3,H,TREF)"

   NPART    = N3 / 3
   IOF1    = NPART
   IOF2    = 2 * NPART
   TSCALE    = 16.0 / (1.0 * NPART - 1.0)

   FOR   I=1 TO N3 STEP 2

1     U1   = Rnd(4711)
     U2   = Rnd(4711)

       V1   =2.0 * U1 - 1.0
       V2   =2.0 * U2 - 1.0
       S     =V1 * V1 +V2 *V2

       IF   ( S >= 1.0 )THEN  GOTO 1
          R    = -2.0 * log (S) / S
          VH(I)  =  V1 * R^.5
          VH(I+1)= V2 * R^.5
          10       NEXT I
 EKIN = 0.0
   SP = 0.0
   FOR    I=1 TO NPART
      SP =SP + VH (I)
20    NEXT I
   SP = SP / NPART
   FOR    I=1 TO NPART
         VH (I) = VH (I) - SP
         EKIN  = EKIN + VH (I) * VH (I)
21    NEXT I
```

```
    SP = 0.0
   FOR    I=IOF1 + 1 TO IOF2
      SP = SP + VH (I)
22   NEXT I
    SP = SP / NPART
    FOR    I=IOF1 + 1 TO IOF2
         VH(I) = VH (I) - SP
         EKIN =  EKIN + VH(I) * VH(I)
23      NEXT I
    SP = 0.0
    FOR  I=IOF2 + 1 TO N3
     SP   = SP + VH (I)
24    NEXT I
    SP = SP / NPART
    FOR  I=IOF2 + 1 TO N3
         VH(I)    = VH (I) - SP
         EKIN    = EKIN + VH (I) * VH (I)
25    NEXT I
    TS = TSCALE * EKIN
    SC = TREF / TS
    SC = (SC)^0.5
    SC = SC * H
    FOR     I=1 TO N3
         VH (I) = VH (I) * SC
         PRINT"VH(";I;")=",VH(I)
30    NEXT I
     END
    DO until I=N3
      F(I)=0.0
    loop
```

-Hasil Eksekusi dari program Maxwell-Boltzmann-1

```
:
VH(1)=        0.20989832e-2
VH(2)=        -0.12618094e-1
VH(3)=        -0.82723833e-2
VH(4)=        -0.10924963e-2
VH(5)=        0.93732508e-2
……………………………..

VH(761)=      -0.99742066e-2
VH(762)=      -0.77111357e-2
VH(763)=      -0.49706784e-2
VH(764)=      0.35381167e-3
VH(765)=      -0.13525497e-1
VH(766)=      0.15105383e-1
VH(767)=      0.10939923e-1
VH(768)=      -0.71259251e-2
```