

MODUL I: PENGANTAR METODE VISUALISASI

Pendahuluan

Pada modul ini anda akan diperkenalkan dengan berbagai metode visualisasi struktur statis dan struktur dinamis dari suatu molekul. Pengertian struktur statis diberikan pada struktur molekul yang diperoleh dari hasil penentuan struktur dengan metode difraksi sinar-X dan NMR, atau hasil dari pembuatan sebuah model. Sedangkan struktur dinamis merujuk pada struktur hasil simulasi dinamika molekul atau monte carlo.

Visualisasi adalah salah satu metode yang sering digunakan untuk memperjelas suatu analisis. Dengan bantuan visualisasi, maka deskripsi suatu analisis akan lebih mudah dilakukan karena bisa langsung merujuk pada bagian-bagian tertentu dari gambar visual suatu objek yang dideskripsikan tersebut.

Pada praktikum kali ini, anda akan diperkenalkan dengan program visual yang dikembangkan oleh Biophysical Theoretical Group dari Universitas Illinois, yaitu Visual Molecular Dynamics (VMD). Program visual ini, memiliki banyak paket visualisasi dan diperkuat dengan scripting language dengan bahasa TCL dan python, tetapi yang utama adalah TCL, untuk membantu mengeksplorasi tidak hanya struktur tetapi aspek fisik, seperti jari-jari girasi, solvent accesible surface area (SASA), dll, dan juga aspek energi seperti energi konformasi (bond, angle, dihedral, torsion), dan energi non-bonding (van der Waals, electrostatic).

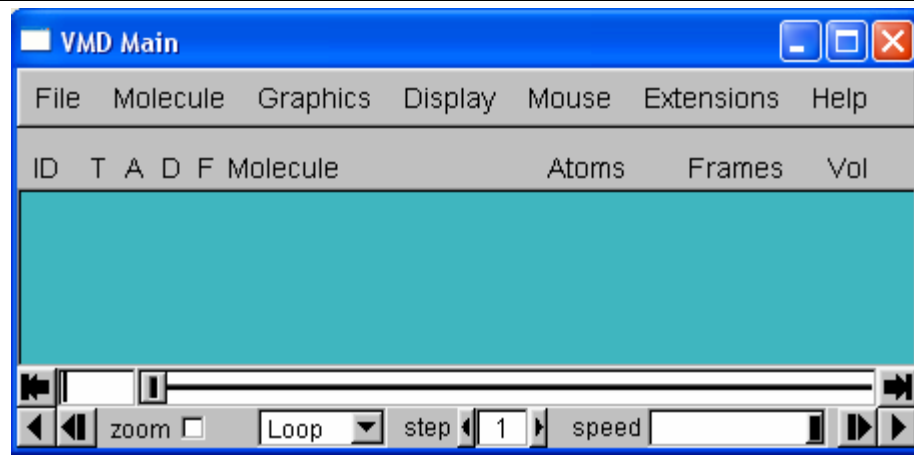
Percobaan kali ini akan dibagi ke dalam dua bagian. Bagian pertama akan difokuskan pada eksplorasi VMD untuk visualisasi struktur dengan mencoba berbagai tipe representasi, dan bagian kedua akan difokuskan pada metode seleksi dan pengantar bahasa TCL. Diakhir praktikum ini, mahasiswa diharapkan dapat memanfaatkan VMD untuk berbagai keperluan analisis yang berkaitan dengan struktur molekul.

Bagian I: Eksplorasi metode visualisasi VMD

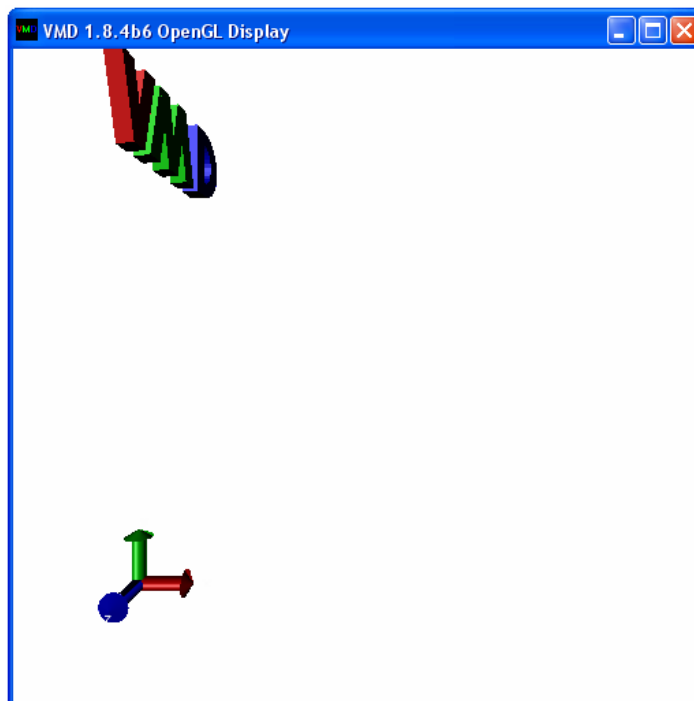
A. Komponen utama VMD

Pada percobaan kali ini akan digunakan VMD versi 1.3 atau 1.4 baik versi windows atau Linux.

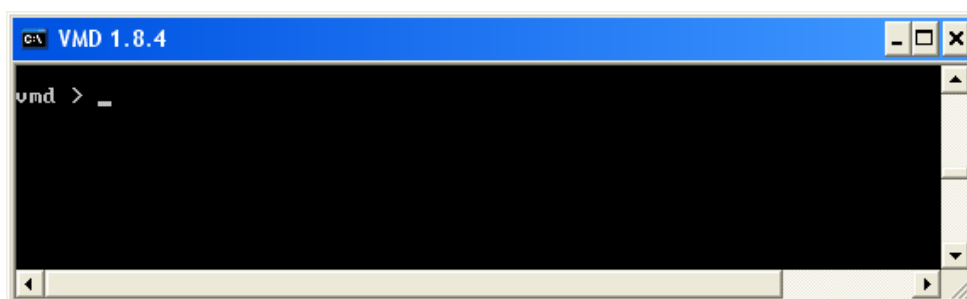
Bila program VMD dijalankan akan muncul tiga windows, yaitu windows utama (Gambar 1), windows openGL (Gambar 2), dan windows console (Gambar 3).



Gambar 1. Windows utama. Windows utama memiliki komponen-komponen kontrol untuk visualisasi dan analisis.



Gambar 2. Windows OpenGL. Windows ini berfungsi untuk mevisualkan struktur.

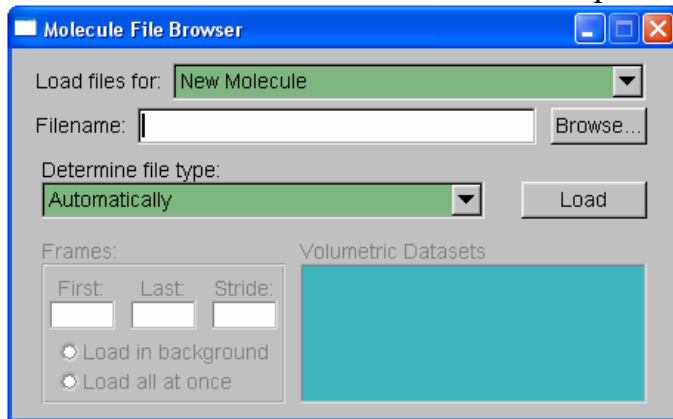


Gambar 3. Windows console. Windows ini dapat digunakan untuk melakukan pengendalian visualisasi dan berbagai analisis berbasis teks.

B. Eksplorasi berbagai representasi

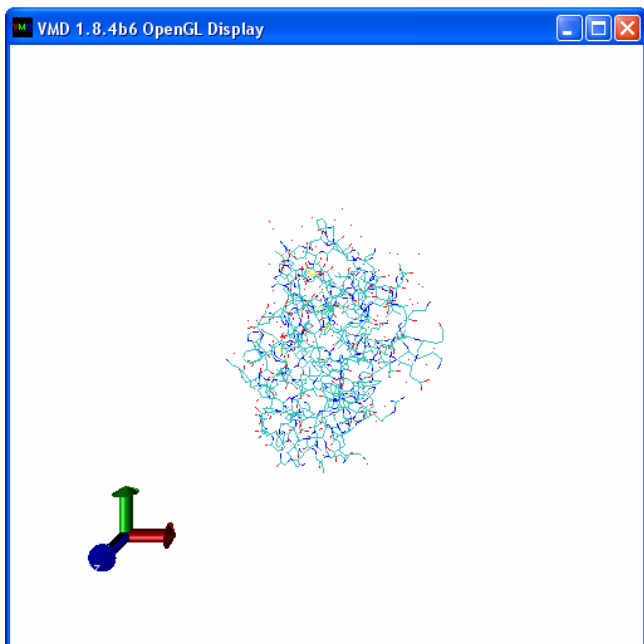
Prosedur:

1. Pada direktori kerja, sudah disiapkan direktori pdb yang berisi beberapa file koordinat biomolekul dengan format pdb (format diberikan pada bagian apendiks) yang didownload dari situs www.rscb.org.
2. Buka vmd, dan pada windows console pindahkan ke direktori pdb berada.
3. Untuk menampilkan struktur suatu molekul, pada windows utama (VMD main), klik File > New Molecule. Anda akan mendapati tampilan windows seperti pada Gambar 4.



Gambar 4.

Kemudian klik tombol Browser dan pilih file mbco.pdb. Selanjutnya tekan tombol load untuk menampilkan file tersebut.



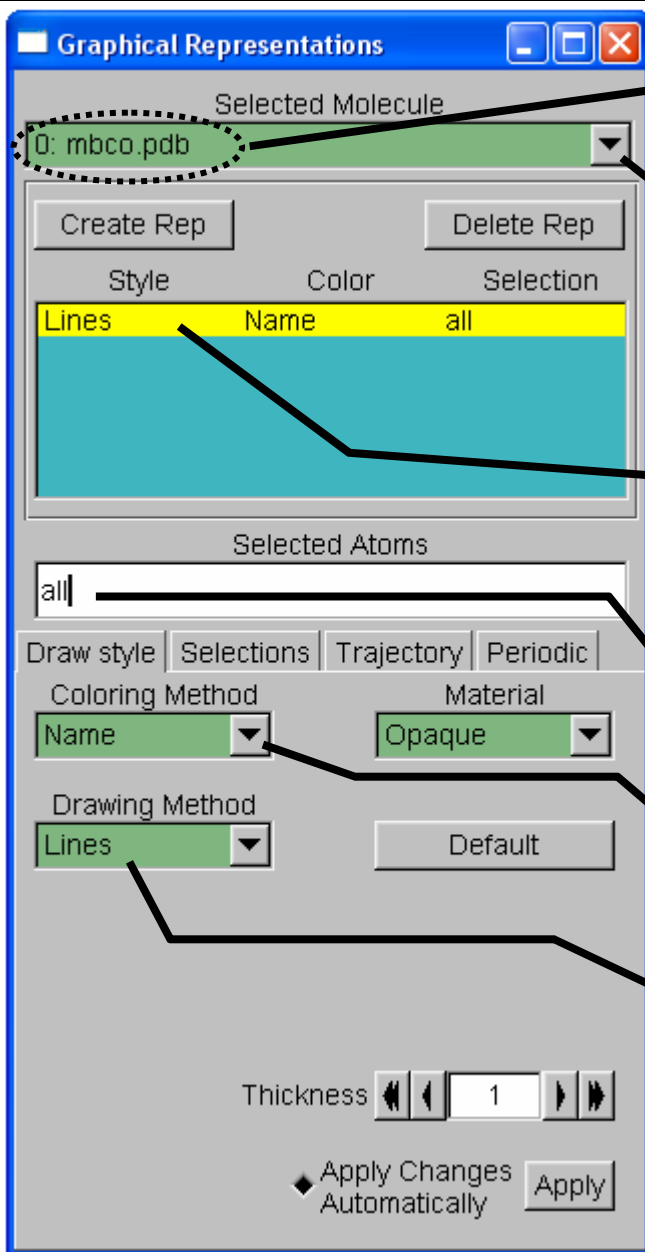
Struktur file mbco.pdb akan muncul pada windows OpenGL (Gambar 5).

Representasi pertama yang muncul adalah tipe garis.

File mbco.pdb berisi koordinat struktur protein mioglobin yang berinteraksi dengan gas CO pada cincin forpirin.

Gambar 5. Representasi standar.

4. Untuk mengubah representasi, klik menu Graphics > Representations, maka akan muncul windows Graphical Representation.



Molecule aktif. VMD memberi angka 0 sebagai nomor urut pertama

Bila lebih dari satu molekul dimuat ke dalam VMD, klik tombol ini untuk memilih molekul yang akan dibuah representasinya.

Representasi yang aktif. Representasi aktif dapat dibuat menjadi beberapa tipe representasi dengan menekan tombol create Rep atau dikurangi dengan menekan tombol Delete Rep

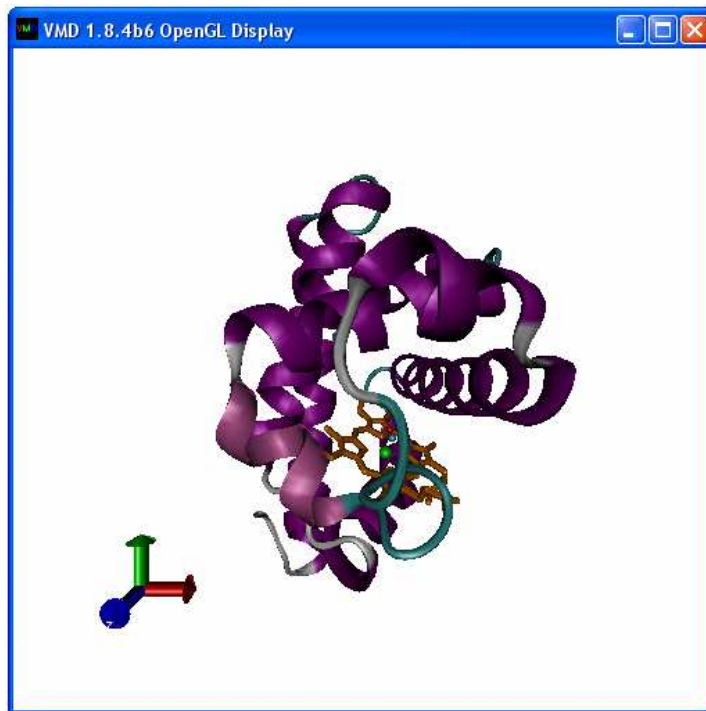
Bagian molekul yang akan diubah representasinya dapat dipilih dengan metode seleksi atom.

Metode pewarnaan (Tabel 1)

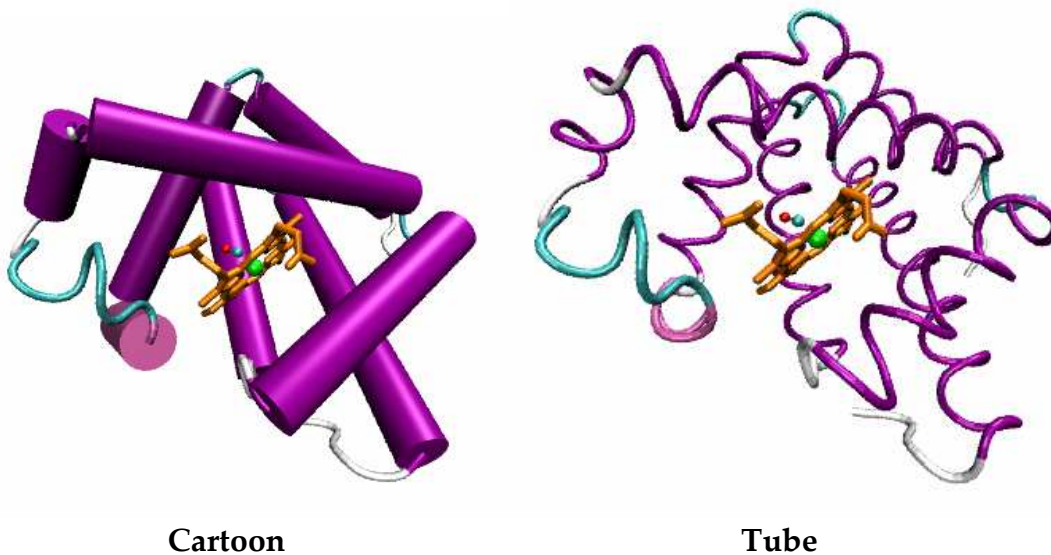
Metode representasi (Tabel 2)

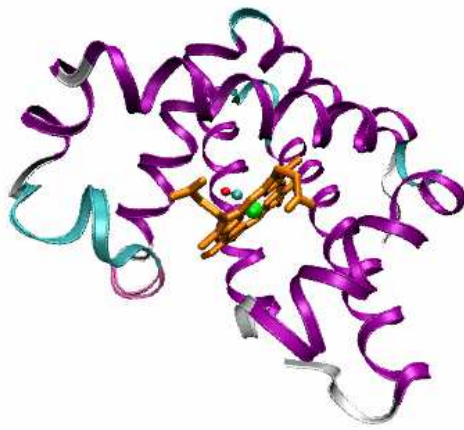
5. Hapus kata all pada kotak Selected Atoms, dan ketikkan kata protein ke dalam kotak tersebut lalu tekan enter. Ubah representasi dengan mengklik panah pada Drawing method dan pilih NewCartoon. Dan ubah metode pewarnaan menjadi structure dengan mengklik tombol panah pada Coloring method.
6. Klik tombol Create Rep. Hapus semua keywords pada kotak Selected atoms dan ketik kata "**resname HEM and not (name FE)**" lalu tekan enter. Pilih representasi "Licorice" pada menu Drawing Method. Selanjutnya ubah warna menjadi "Orange" pada menu Coloring Method.
7. Klik tombol Create Rep, dan hapus seluruh keywords pada kotak Selected atoms dan ketik kata "**name FE**". Pilih representasi "VDW" pada menu Drawing Method. Ubah Sphere Scale menjadi 0,5 dan Sphere Resolution menjadi 10.
8. Klik tombol Create Rep sekali lagi, dan ganti keywords pada kotak Selected atoms dengan keywords "**resname CO**". Pilih representasi "CPK" pada menu Drawing Method untuk menggambarkan struktur molekul ini.

9. Pada windows console, ketikkan perintah **“scale by 1.5”** lalu enter. Ukuran gambar pada windows OpenGL akan membesar 1,5 kali gambar sebelumnya. Tampilan akhirnya akan nampak seperti pada gambar dibawah ini.

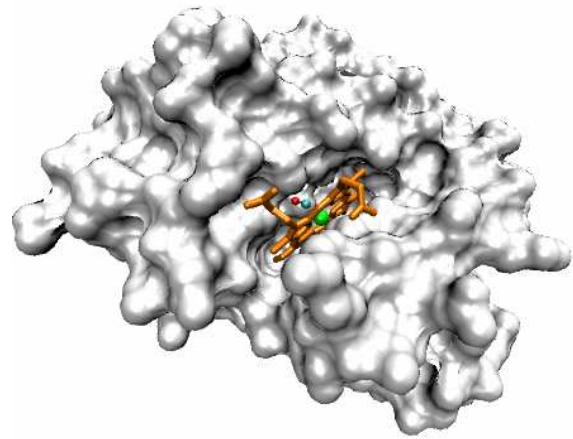


10. Coba ubah tipe representasi untuk protein menjadi **“Cartoon”**, **“Tube”**, **“Ribbons”** atau **“Surf”**. Gambar struktur yang dihasilkan akan seperti pada gambar di bawah ini.





Ribbons



Surf

Eksplorasi struktur

Untuk dapat memanfaatkan VMD dalam mengeksplorasi struktur, maka metode seleksi yang berlaku pada VMD harus dikuasai. Metode seleksi pada VMD diadaptasi dari bahasa TCL. Kata-kata kunci (keywords) yang tersedia dalam VMD yang digunakan dalam metode seleksi diberikan pada Tabel 3. Bila definisi kata kunci diikuti oleh *bool* (boolean), maka kata kunci tersebut dapat memiliki nilai "on" atau "off". Bila diikuti oleh *str*, maka kata kunci tersebut mengambil nilai dalam konteks huruf. Bila diikuti dengan *num*, maka kata kunci mengambil nilai dalam konteks angka.

A. Katakunci Boolean

Kata kunci boolean dapat berdiri sendiri atau dapat diikuti dengan parameter logika "or", "and", dan "not".

Contoh:

- **protein** → hanya memilih atom-atom protein
- **water** → hanya memilih atom-atom air
- **protein and backbone** → hanya memilih atom-atom backbone protein
- **resname HEM and not (name FE)** → hanya memilih residue HEM tetapi tidak termasuk atom FE
- **resid 50 and resid 70** → hanya menampilkan residu nomor 50 dan 70

Bila dua kata kunci digunakan tanpa menggunakan parameter logika, maka parameter logika "and" akan disisipkan secara implisit.

Contoh:

- **protein backbone** → ekuivalen dengan **protein and backbone**
- **resid 50 70** → ekuivalen dengan **resid 50 and resid 70**

B. within dan exwithin

Kata kunci "within" berarti "cari atom A yang berada pada jarak tertentu dari atom B". Contoh:

- **protein within 5 of resname HEM** → Cari atom-atom protein yang berada pada jarak 5 Å dari residu bernama HEM.

Kata kunci "exwithin" berarti hilangkan atom A yang berada diluar jarak tertentu dari atom B". Contoh:

- **all exwithin 5 of resname HEM**

C. within and same

VMD juga menyediakan metode seleksi kombinasi antara within dan same, yaitu:

within <number> of <selection> and same <keyword> as <selection>.

Bagian "within" memilih seluruh atom yang berada pada jarak tertentu (dalam Å) dan bagian "same" mempersempit pemilihan hanya untuk atom dengan kata kunci yang sama.

Contoh:

- **within 5 of protein and same residue as water** → Memilih residu air yang berada pada jarak 5 Å dari protein.
- **same fragment as resid 35** → "same" juga dapat berdiri sendiri. Contoh ini berarti dicari fragmen suatu molekul yang memiliki residu yang sama dengan residu no 35.

D. Mengapit dengan tanda petik tunggal dan ganda

Fungsi dari tanda petik tunggal pada VMD adalah untuk memberikan spasi atau untuk menyisipkan karakter alpha numerik. Sebagai contoh beberapa molekul memiliki nama residu dengan spasi di antaranya.

Contoh:

- **resname 'A 1'**
- **name '05*'** → Bila tidak diberi tanda petik tunggal, maka karakter * akan disamakan dengan perkalian.

Fungsi dari tanda petik ganda adalah untuk menjalankan ekspresi. Sebagai contoh:

- **name "C.*"** → memilih atom dengan diawali huruf C
- **resname "A.*" GLY ".*T"** → memilih atom dengan nama residu diawali dengan huruf A, kemudian memilih residu Gly, dan memilih residu dengan diakhiri dengan huruf T.

Bila memilih atom yang mengandung karakter alpha numerik seperti +, -, atau *, maka tanda \ (backslash) harus disisipkan sebelum karakter alpha-numerik. Contoh:

- **name "Na\+"** → memilih atom dengan nama Na⁺.

Berbagai metode seleksi atom berikut memiliki pengertian yang sama dengan **name CA CB**

- **name "CA|CB"**
- **name "C[AB]"**
- **name "C(A|B)"**

E. Seleksi komparasi

Tanda $<$, $>$, $<=$, $=$, $>=$, $!=$ dapat digunakan dalam metode seleksi. Contoh:

- **mass > 5** → memilih atom dengan massa > 5 s.m.a (satuan massa atom)
- **sqr(x-3)+sqr(y-4)+sqr(z+5) <= sqr(5)** → memilih atom dengan jarak ≤ 5 Å dari koordinat (3, 4, -5)

F. Sequence

Katak kunci "sequence" digunakan untuk memilih atom berdasarkan urutan yang diberikan setelah kata kunci ini. Contoh:

- **sequence APD** → Memilih asam amino Ala, Phe, Asp
- **sequence AATCGGAT**

Menyimpan hasil kerja

- Dalam bentuk formasi terakhir: File > Save State
- Dalam bentuk file gambar: File > Render

PRAKTIKUM VISUALISASI**Praktikum I**

1. Muat file pdb mbco.pdb ke layar window OpenGL VMD. Ubah tampilan sesuai dengan daftar dibawah ini:
 - a. Protein : Surf (transparan) dan NewCartoon
 - b. Gugus HEM tanpa FE: Licorice
 - c. Atom FE: VDW sesuaikan ukuran jari-jari atom agar nampak tidak terlalu besar
 - d. Molekul CO: VDW
 - e. Warna: Surf = putih; NewCartoon = structure; Gugus HEM, FE & CO: Name.
 - f. Render hasilnya dengan menggunakan Tachyon dan berinama file output dengan kode NIM_mbco.bmp.
 - g. Simpan tampilan pertama dengan menggunakan Save State dengan nama NIM_mbco.vmd
2. Hapus representasi Surf, kemudian lakukan metode seleksi untuk mencari asam amino yang ikut berperan dalam pembentukan kompleks dengan ion Fe^{2+} dan tampilkan asam amino tersebut dengan tampilan CPK dengan metode pewarnaan "Name".
3. Lakukan pula metode seleksi untuk mencari asam-asam amino yang berperan dalam menstabilkan posisi residu HEM. Untuk melakukan analisis ini, persempit pencarian dengan hanya menampilkan residu-residu yang dekat dengan gugus HEM.
4. Setelah residu asam amino yang berperan dalam stabilitas dan pembentukan kompleks diketahui, gunakan metode seleksi untuk menampilkan hanya residu-residu tersebut dan gunakan representasi CPK atau licorice. Metode pewarnaan harus disesuaikan sehingga identitas residu-residu yang berperan tersebut dengan mudah dapat dibedakan satu sama lain.
5. Simpan hasilnya dalam file NIM_active_site.vmd dan render hasilnya dengan Tachyon dengan nama NIM_active-ste.vmd.

Praktikum II

1. Buka file 1CA2.pdb (Enzim carbonic anhydrase).
2. Tampilkan komponen protein dengan representasi "New Cartoon" dan atom ZN dengan representasi "VDW". Gunakan metode pewarnaan "structure" untuk protein dan "name" untuk atom ZN. Render hasilnya dengan metode Tachyon dan simpan dalam file NIM_CA2.bmp.
3. Enzim CA2 memiliki ion Zn^{2+} , gunakan metode seleksi untuk mencari asam-asam amino dan molekul air yang terlibat dalam pembentukan kompleks. Apa bentuk kompleks antara CA2 dengan residu-residu tersebut.
4. Tampilkan residu-residu hasil penentuan pada no. 3 dengan representasi "Licorice", pilih warna yang sesuai. Render hasilnya dengan metode Tachyon dan simpan dalam file NIM_CA2_activesite.bmp
5. Gunakan metode seleksi untuk menampilkan asam amino bermuatan negatif dan positif.
6. Diskusikan dengan asisten cara-cara yang lebih praktis untuk menemukan residu-residu yang terlibat dalam pembentukan jembatan garam dalam molekul CA2.
7. Bila pasangan-pasangan residu yang membentuk jembatan garam telah ditemukan, tampilkan residu-residu tersebut dengan metode representasi "Licorice" dan metode pewarnaan "Restype". Buat representasi "New Cartoon" untuk protein menjadi transparan sehingga dapat diketahui dengan jelas seluruh jembatan garam yang terdapat dalam molekul CA2.
8. Render hasilnya dan simpan dalam file NIM_CA2_salt_bridge.bmp

Tabel 1. Metode pewarnaan

Method	Deskripsi
Name	Atom name, using the Name category
Type	Atom type, using the Type category
ResName	Residue name, using the Resname category
ResType	Residue type, using the Restype category
ResID	Residue identifier, using the resid mod 16 for the color
Chain	The one-character chain identifier, using the Chain category
SegName	Segment name, using the Segname category
Molecule	Molecule all one color, using the Molecule category
Structure	Helix, sheet, and coils are colored differently
ColorID	Use a user-specified color index (from 0 to 15)
Beta	Color scale based on beta value of the PDB file
Occupancy	Color scale based on the occupancy field of the PDB file
Mass	Color scale based on the atomic mass
Charge	Color scale based on the atomic charge
Pos	Color scale based on the distance of each atom to the center of the molecule. This is an interesting way to view globular systems.
User	Provides a single data value for each atom for each timestep
Index	Color scale is based on the atom index. Due to the way the PDB file is organized, this actually looks nice.
Backbone	Backbone atoms green, everything else is blue
Throb	Animates colors through the active color scale based on wall clock time
Timestep	Color scale based on the current trajectory timestep frame

Tabel 2. Tipe representasi

Metode	Deskripsi
Lines	simple lines for bonds, points for atoms
Bonds	lighted cylinders for bonds
DynamicBonds	dynamically calculated distance-based bonds
HBonds	display hydrogen bonds
Points	just points for atoms, no bonds
VDW	solid van der Waal spheres for atoms, no bonds
CPK	scaled VDW spheres, with cylinders for bonds
Licorice	spheres for atoms, cylinders for bonds, same radius
Trace	connected cylindrical segments through C α atoms
Tube	smooth cylindrical tube through the C α atoms
Ribbons	flat ribbon through the C α atoms
NewRibbons	smooth ribbon through the C α atoms
Cartoon	cartoon diagram (cylinders and ribbons) based on secondary structure
NewCartoon	cartoon diagram (spiral and tube) based on secondary structure
MSMS	molecular surface as determined by the program MSMS
Surf	molecular surface as determined by SURF
VolumeSlice	display a texture mapped slice from a volumetric data set
Isosurface	display an isovalue surface from a volumetric data set
Dotted	dotted van der Wall spheres for atoms, no bonds
Solvent	dotted representation of the solvent accessible surface

Tabel 3. Kata kunci dalam metode seleksi

Keywords	Arg	Description
all	<i>bool</i>	everything
none	<i>bool</i>	nothing
name	<i>str</i>	atom name
type	<i>str</i>	atom type
index	<i>num</i>	the atom number, starting at 0
serial	<i>num</i>	the atom number, starting at 1
atomicnumber	<i>num</i>	atomic number (0 if undefined)
element	<i>str</i>	atomic element symbol string ('X' if undefined)
altloc	<i>str</i>	alternate location/conformation identifier
chain	<i>str</i>	the one-character chain identifier
residue	<i>num</i>	a set of connected atoms with the same residue number
protein	<i>bool</i>	a residue with atoms named C, N, CA, and O
nucleic	<i>bool</i>	a residue with atoms named P, O1P, O2P and either O3', C3', C4', C5', O5' or O3*, C3*, C4*, C5*, O5*. This definition assumes that the base is phosphorylated,
backbone	<i>bool</i>	an assumption which will be corrected in the future.
sidechain	<i>bool</i>	the C, N, CA, and O atoms of a protein and the equivalent atoms in a nucleic acid.
water,	<i>bool</i>	
waters	<i>bool</i>	non-backbone atoms and bonds
fragment	<i>num</i>	all atoms with the resname H2O, HH0, OHH, HOH,
pfrag	<i>num</i>	OH2, SOL, WAT, TIP, TIP2, TIP3 or TIP4
nfrag	<i>num</i>	a set of connected residues
sequence	<i>str</i>	a set of connected protein residues
numbonds	<i>num</i>	a set of connected nucleic residues
resname	<i>str</i>	a sequence given by one letter names
resid	<i>num</i>	number of bonds
segname	<i>str</i>	residue name
x, y, z	<i>float</i>	residue id
radius	<i>float</i>	segment name
mass	<i>float</i>	x, y, or z coordinates
charge	<i>float</i>	atomic radius
beta	<i>float</i>	atomic mass
occupancy	<i>float</i>	atomic charge
user	<i>float</i>	temperature factor
at	<i>bool</i>	occupancy
acidic	<i>bool</i>	time-varying user-specified value
acyclic	<i>bool</i>	residues named ADA A THY T
aliphatic	<i>bool</i>	residues named ASP GLU
alpha	<i>bool</i>	"protein and not cyclic"
amino	<i>bool</i>	residues named ALA GLY ILE LEU VAL
aromatic	<i>bool</i>	atom's residue is an alpha helix
basic	<i>bool</i>	a residue with atoms named C, N, CA, and O
bonded	<i>bool</i>	residues named HIS PHE TRP TYR
buried	<i>bool</i>	residues named ARG HIS LYS
cg	<i>bool</i>	atoms for which numbonds > 0
charged	<i>bool</i>	residues named ALA LEU VAL ILE PHE CYS MET TRP
cyclic	<i>bool</i>	residues named CYT C GUA G
hetero	<i>bool</i>	"basic or acidic"

hydrogen	<i>bool</i>	residues named HIS PHE PRO TRP TYR
large	<i>bool</i>	"not (protein or nucleic)"
medium	<i>bool</i>	name "[0-9]?H.*"
neutral	<i>bool</i>	"protein and not (small or medium)"
polar	<i>bool</i>	residues named VAL THR ASP ASN PRO CYS ASX PCA HYP
purine	<i>bool</i>	residues named VAL PHE GLN TYR HIS CYS MET TRP ASX GLX PCA HYP
pyrimidine	<i>bool</i>	"protein and not hydrophobic"
small	<i>bool</i>	residues named ADE A GUA G
surface	<i>bool</i>	residues named CYT C THY T URI U
helix	<i>bool</i>	residues named ALA GLY SER
helix_3_10	<i>bool</i>	"protein and not buried"
extended_beta	<i>bool</i>	atom's residue is an alpha helix
bridge_beta	<i>bool</i>	atom's residue is an alpha helix
rasmol	<i>str</i>	atom's residue is a beta sheet
alpha_helix	<i>bool</i>	atom's residue is a beta sheet
pi_helix	<i>bool</i>	translates Rasmol selection string to VMD
helix	<i>bool</i>	atom's residue is an alpha helix
sheet	<i>bool</i>	atom's residue is a pi helix
turn	<i>bool</i>	atom's residue is an alpha or pi helix
coil	<i>bool</i>	atom's residue is a beta sheet
structure	<i>str</i>	atom's residue is in a turn conformation
phi, psi	<i>float</i>	atom's residue is in a coil conformation
within	<i>str</i>	single letter name for the secondary structure
exwithin	<i>str</i>	backbone conformational angles
same	<i>str</i>	selects atoms within a specified distance of a selection (i.e within 5 of name FE).
ufx, ufy, ufz	<i>num</i>	exclusive within, equivalent to (within 3 of X) and not X. selects atoms which have the same keyword as the atoms in a given selection (i.e. same segname as resid 35) force to apply in the x, y, or z coordinates

Pengantar bahas Tcl

Pendahuluan

Tcl adalah singkatan dari Tool Command Language yang umumnya dibaca sebagai “tickle” diciptakan pertama kali oleh John Ousterhout di tahun 1988. Tcl adalah suatu “interpreted programming language” yang memiliki berbagai fasilitas yang umumnya ada pada bahasa pemrograman lain seperti variabel, prosedur, dan struktur kontrol dan juga fitur-fitur lain yang tidak ditemui di kebanyakan bahasa pemrograman. Tcl dapat berjalan di hampir seluruh sistem operasi seperti Unix, Macintosh, dan Windows. Kelebihan lain dari Tcl adalah dalam hal kefleksibelannya. Tcl banyak digunakan untuk menginteraksikan berbagai program eksternal, berperan sebagai library program-program aplikasi, dan juga sebagai “scripting language”.

Pada modul ini akan diberikan dasar-dasar pemrograman dengan bahasa Tcl dan beberapa aplikasinya untuk eksplorasi struktur dan analisis trajektori simulasi.

Sumber

Untuk versi Unix/Linux, Tcl dapat diperoleh di <http://tcl.sourceforge.net/>, sedangkan untuk versi windows dapat diperoleh di <http://www.activestate.com/Products/ActiveTcl/>

Untuk saat ini, akan digunakan Tcl versi 8.4 untuk windows dari activestate.

tclsh84

Paket program tcl8.4 dari activestate menyediakan tiga konsol untuk keperluan pemrograman, yaitu tclsh84, tkcon, dan wish84. Untuk pemrograman tcl digunakan tclsh84, sedangkan untuk pemrograman aplikasi grafik digunakan tkcon dan wish84. Untuk saat ini, kita hanya akan menggunakan konsol tclsh84.

Apabila konsol tclsh84 dibuka akan muncul windows dengan “prompt” %. Keluarnya tanda % menandakan tclsh siap menerima perintah-perintah berbahasa tcl.

Pemrograman interaktif