

FILE EKSTERNAL DALAM PROLOG

File Eksternal

File eksternal adalah media penyimpanan nilai-nilai pada tempat penyimpanan sekunder. Karena menyangkut tempat penyimpanan (memori), maka merupakan aspek prosedural.

Program deklaratif tidak mengenal tempat penyimpanan karena program terdiri dari fakta serta aturan, dan pengguna melakukan Query. Semua fakta dan aturan tersimpan di dalam program, dan biasanya Query diajukan ke program secara interaktif. Maka program deklaratif murni tidak memerlukan file eksternal. Namun seringkali list yang dimanipulasi sebagai objek internal dalam program deklaratif dikehendaki bukan sebagai “konstanta” di dalam program, atau jika merupakan “hasil” masih dapat diperoleh kembali karena “disimpan”. Maka beberapa program deklaratif memerlukan file eksternal untuk tempat penyimpanan sehingga dapat diakses kembali.

Eksternal Sequential File dalam PROLOG

Sintaks :

Prolog	Turbo Prolog	Penjelasan
see (FileInput)	openread>NamaLojik, NamaFisik)	Membuka file input dengan nama logik>NamaLojik dan nama fisik>NamaFisik.
seeing(CInput)	readdevice>NamaLojik)	Memindahkan 'current input device' ke>NamaLojik.
seen	closefile>NamaLojik), readdevice(keyboard)	Menutup file input dan sekaligus memindahkan 'current input device' ke keyboard.
tell(FileOutput)	openwrite>NamaLojik, NamaFisik)	Membuka file output dengan nama logik>NamaLojik dan nama fisik>NamaFisik.
telling(COutput)	writedevic>NamaLojik)	Memindahkan 'current output device' ke>NamaLojik.
told	closefile>NamaLojik), writedevic(screen)	Menutup file output dan sekaligus memindahkan 'current output device' ke screen.

Contoh :

Notasi Logik	Notasi Turbo Prolog
BACA FILE-TEKS Kamus dtread : string fin : textfile Algoritma assign(fin,'input.txt') reset(fin) <u>while</u> not eof(fin) <u>do</u> <input (fin,dtread)<br=""/> output (dtread) close(fin)	BACA FILE-TEKS Domains file = fin Predicates go Goal /* internal */ go Clauses go :- openread(fin,"input.txt"), readdevice(fin), while(fin), closefile(fin), readdevice (keyboard) . /* kondisi berulang */ while(FInput) :- not (eof(FInput)), readln(Dtread), write(Dtread), nl, while(FInput) . /* kondisi berhenti */ while(FInput)) :- eof(FInput) .
TULIS KE FILE-TEKS Kamus i : <u>integer</u> fout : textfile Algoritma assign(fout,' <u>output.txt</u> ') rewrite(fout) i ← 1 <u>while</u> i <= 10 <u>do</u> <u>output</u> (fout,i) i ← i+1 close(fout)	TULIS KE FILE-TEKS Domains dindeks = integer file = fout Predicates go while(dindeks,dindeks,dindeks) Goal /* internal */ go Clauses go :- openwrite(fout,"output.txt"), writedevise(fout), I = 1, while(I,1,10), closefile(fout), writedevise(screen) . /* kondisi berulang */ while(I,Next,Akhir) :- I <= Akhir, write(I), nl, I1 = I + Next, while(I1,Next,Akhir) . /* kondisi berhenti */ while(I,_,Akhir) :- I > Akhir.

(lanjutan Eksternal Sequential File)

Notasi Logik	Notasi Turbo Prolog
<p>BACA FILE-TEKS, TAMPUNG DI LIST</p> <p>Kamus</p> <pre> dtread : string lis : list of char fin : textfile </pre> <p>Algoritma</p> <pre> assign(fin, 'input.txt') reset(fin) lis ← nil while not eof(fin) do input(fin, dtread) lis ← konso(dtread, lis) lis ← inverse(lis) close(fin) </pre>	<p>BACA FILE-TEKS, TAMPUNG DI LIST</p> <p>Domains</p> <pre> dlist = delemen* delemen = string file = fin </pre> <p>Predicates</p> <pre> go while(file, dlist) </pre> <p>Goal /* internal */</p> <pre> go </pre> <p>Clauses</p> <pre> go :- openread(fin, "input.txt"), readdevice(fin), Lis = [], while(fin, Lis), closefile(fin), readdevice(keyboard). /* kondisi berulang */ while(FInput, Ls) :- not(eof(FInput)), readln(Dtread), write(Dtread), nl, while(FInput, [Dtread Ls]). /* kondisi berhenti */ while(FInput, _) :- eof(FInput). </pre>
<p>DR KEYBOARD, TAMPUNG DI LIST</p> <p>Kamus</p> <pre> dtread:string lis : list of char </pre> <p>Algoritma</p> <pre> lis ← nil input(dtread) while not (dtread = '999') do lis ← konso(dtread, lis) input(dtread) lis ← inverse(lis) </pre>	<p>DARI KEYBOARD, TAMPUNG DI LIST</p> <p>Domains</p> <pre> dlist = delemen* delemen = string </pre> <p>Predicates</p> <pre> go while(delemen, delemen, dlist) </pre> <p>Goal /* internal */</p> <pre> go </pre> <p>Clauses</p> <pre> go :- Lis = [], readln(Dtread), while(Dtread, "999", Lis). while(Dtread, Akhir, Ls) :- not(Dtread = Akhir), Ls1 = [Dtread Ls], readln(Dtread), while(Dtread, Akhir, Ls1). while(Akhir, Akhir, _). </pre>

(lanjutan Eksternal Sequential File)

Notasi Logik	Notasi Turbo Prolog
BACA FILE-OF-INTEGER, TAMPUNG DI LIST Kamus i : <u>integer</u> lis : list of <u>integer</u> fin : file of <u>integer</u> Algoritma assign(fin, ' <u>input.int</u> ') reset(fin) lis ← nil <u>while</u> not eof(fin) <u>do</u> <u>input</u> (fin, i) lis ← konso(i, lis) lis ← inverse(lis) close(fin)	BACA FILE-OF-INTEGER, TAMPUNG DI LIST Domains dlist = delemen* delemen = integer file = fin Predicates go while(file, dlist) Goal /* internal */ go Clauses go :- openread(fin, "input.int"), readdevice(fin), Lis = [], while(fin, Lis), closefile(fin), readdevice(keyboard). /* kondisi berulang */ while(FInput, Ls) :- not(eof(FInput)), readint(Dtread), Ls1 = [Dtread Ls], while(FInput, Ls1). /* kondisi berhenti */ while(FInput, _) :- eof(FInput).

BASIS DATA

DALAM PROLOG

Pengertian Basis Data

Program PROLOG secara konseptual merupakan deskripsi dari sebuah persoalan dunia nyata.

Sehingga program PROLOG dapat dipandang sebagai sebuah basisdata.

Dalam hal ini, pengertian basisdata adalah sebuah program yang terdiri dari sejumlah besar fakta dan beberapa aturan.

Fakta merepresentasikan data yang eksplisit, sedangkan aturan merepresentasikan informasi yang implisit.

Basisdata Internal dalam Turbo PROLOG

Pada konteks bahasa Turbo PROLOG terdapat beberapa cara untuk merepresentasikan basisdata:

1. Basisdata Statik

Merupakan kumpulan fakta yang menjadi **bagian dari program**, sehingga sifatnya statik.

Jika ada modifikasi terhadap fakta, maka program harus dimodifikasi. Basis data statik disimpan sebagai bagian dari predikat program (PREDICATES).

2. Basisdata Dinamik

Merupakan kumpulan fakta yang disimpan **terpisah dari program** (disimpan dalam file khusus), sehingga sifatnya dinamik. Jika ada modifikasi terhadap fakta, maka program tetap. Yang berubah hanya file basis data. Untuk basisdata dinamik, basisdata dapat disimpan secara internal dalam memori (pada saat pemrosesan) atau secara eksternal. Dalam hal ini yang akan dibahas hanya yang bersifat internal. Basis data dinamik jenis internal disimpan sebagai bagian dari basisdata program (DATABASES).

Pada PROLOG, khususnya Turbo PROLOG, terdapat beberapa predikat sistem yang ditujukan untuk memanipulasi basisdata. Predikat tersebut dijelaskan dalam tabel berikut.

Perintah dan deskripsi ringkas dari parameternya diberikan pada tabel sebagai berikut:

Prolog	Turbo Prolog	Penjelasan
listing(NmPred)		Mendaftarkan semua predikat bernama NmPred.
asserta(Fakta)	asserta(Fakta) asserta(Fakta, NamaBD)	Menambahkan Fakta di bagian awal basis data (bernama NamaBD).
assertz(Fakta)	assertz(Fakta) assertz(Fakta, NamaBD)	Menambahkan Fakta di bagian akhir basis data (bernama NamaBD).
retract(Fakta)	retract(Fakta) retract(Fakta, NamaBD)	Menghapus Fakta dari basis data (bernama NamaBD).
	retractall(Fakta) retractall(Fakta, NamaBD)	Menghapus semua fakta yang dapat berunifikasi dengan Fakta (dari basis data bernama NamaBD).
consult(NamaFile)	consult(NamaFile) consult(NamaFile, NamaBD)	Membaca isi sebuah file bernama NamaFile dan menyimpannya dalam memory sebagai bagian dari basis data (statik, atau dinamik bernama NamaBD).
	save(NamaFile) save(NamaFile, NamaBD)	Menyimpan bagian dari basis data (statik, atau dinamik bernama NamaBD) dari memory ke dalam file bernama NamaFile.