

BAB III

ALGORITMA DAN STRUKTUR DATA

3.1. Pengertian Algoritma

Algoritma adalah suatu konsep dalam pemecahan masalah matematika tertentu, yang memuat instruksi-instruksi dan harus diikuti langkah demi langkah menuju solusinya.

3.2. Struktur Data

Suatu struktur data secara umum terdiri dari beberapa bagian seperti himpunan nilai - nilai data dan sejumlah operasi dasar yang bekerja pada data tersebut menurut suatu algoritma tertentu.

Suatu struktur data mempunyai 3 bagian utama, yaitu :

1. Himpunan struktur dari tempat penyimpanan atau *storage*.

Merupakan koleksi dari variabel dan hubungan antara satu variabel dengan variabel yang lain.

2. Himpunan dari fungsi - fungsi dasar.

Dapat digunakan pada struktur tempat penyimpanan yang ada dan dapat digunakan pada setiap bagian program.

3. Himpunan dari algoritma.

Digunakan untuk pengubahan dari struktur tempat penyimpanan.

3.2.1. Tipe Data Abstrak (TDA)

Definisi : 3.1.

Tipe data sesuatu variabel adalah kumpulan nilai - nilai yang dapat dimuat oleh variabel ini.

Contoh : 3.1.

Sebuah tipe boolean hanya bernilai True atau False, dan tidak boleh berisi nilai yang lain.

Untuk merepresentasikan suatu model matematis dari suatu TDA, digunakan struktur data yang berisi sekumpulan variabel yang dapat terdiri atas beberapa tipe data dan mempunyai bermacam - macam jenis dan cara relasi antara setiap variabel.

Definisi : 3.2.

Satuan dasar pembentuk struktur data berupa sel, dimana setiap sel dapat dibentuk atas gabungan beberapa tipe data dasar atau tipe data abstrak yang lain.

Contoh : 3.2.

Contoh dari definisi ini akan ditunjukkan pada sub bab 3.2.4. mengenai List.

3.2.2. Konsep TDA Linked - List

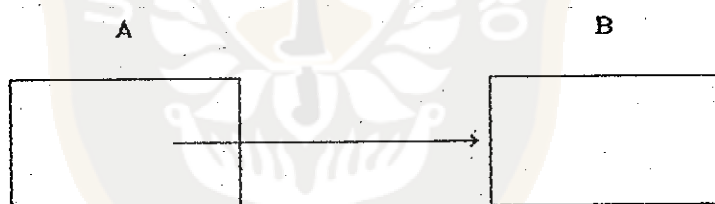
Pada bagian pendahuluan telah disebutkan bahwa dalam tulisan ini penggunaan Tipe Data

Abstrak hanya akan dibatasi pada penggunaan TDA linked - list. (<http://eprints.undip.ac.id>)

Dalam penggunaan TDA linked - list, *pointer* mempunyai peranan yang sangat penting, sebab setiap elemen dalam linked - list selalu berisi pointer. Dalam penggambaran struktur data, pointer digambarkan sebagai anak panah.

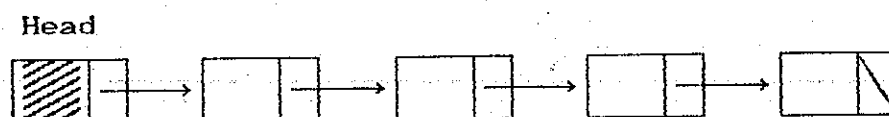
Contoh : 3.3.

Sel A merupakan pointer ke sel B digambarkan sebagai sebuah anak panah dari sel A ke sel B, dimana elemen A dan elemen B dapat berupa data atau sembarang elemen yang merupakan suatu *record* yang dapat berisi pointer juga.



Gambar : 3.1. Pointer dari A ke B

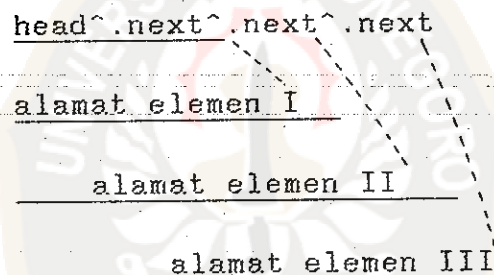
Pada umumnya suatu linked - list memiliki sejumlah elemen, dimana elemen yang paling awal bersifat agak khusus dan tidak digunakan untuk menyimpan data. Elemen paling awal tersebut dikenal dengan nama elemen kepala atau *head*.



Gambar : 3.2. List sederhana dengan sebuah elemen kepala

Alamat elemen kepala list biasanya disimpan pada variabel head, sehingga variabel head tersebut menunjukkan suatu lokasi memori yang terdiri atas 2 field yaitu field *data* dan *next*. Selanjutnya alamat elemen pertama list disimpan pada field *next* elemen itu ($head^{next}$). Jadi $head^{next}$ menunjukkan lokasi memori yang terdiri atas 2 field yaitu *data* dan *next*.

Hal ini dapat digambarkan sebagai berikut :



Prosedur - prosedur dasar yang dapat bekerja

pada struktur linked - list yaitu :

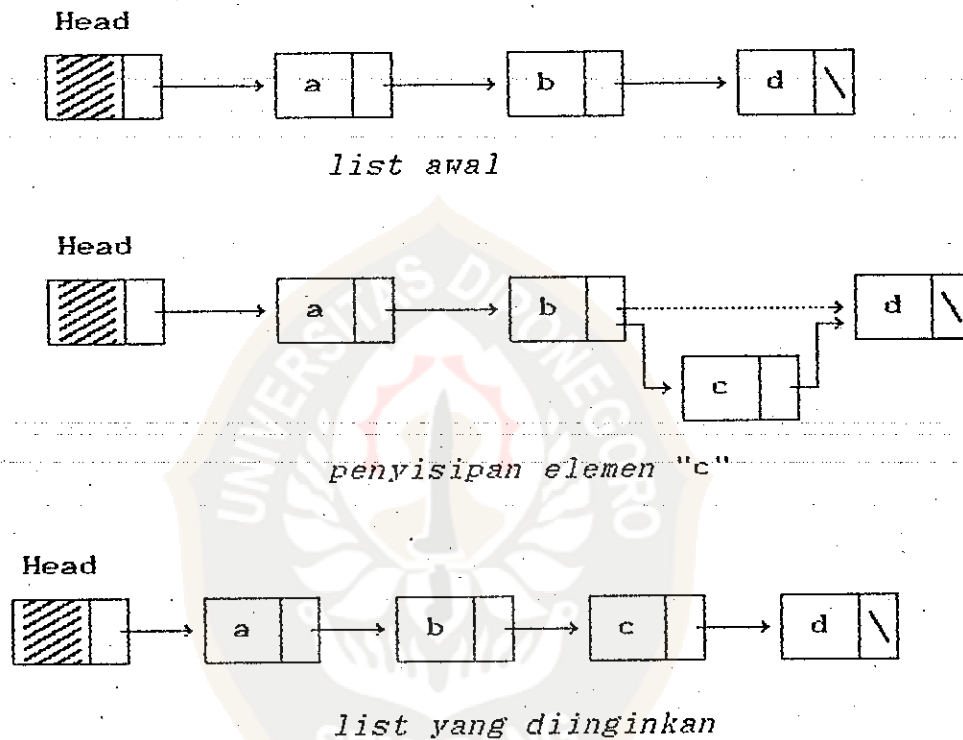
- Mencari lokasi tempat penyisipan.
- Menyisipkan sebuah elemen list.
- Menghapus sebuah elemen list.

Dengan adanya kemudahan dalam hal memperluas atau memperkecil list (dengan jalan menambah atau menghapus elemen), kemudahan untuk mengakses setiap elemen, memodifikasi, atau menyisipkan elemen lain menjadikan struktur elemen ini fleksibel.

Selain itu beberapa list dapat dengan mudah digabungkan menjadi sebuah list baru, sedangkan sebuah list - pun dapat dipisahkan menjadi beberapa sub list.

Contoh : 3.4.

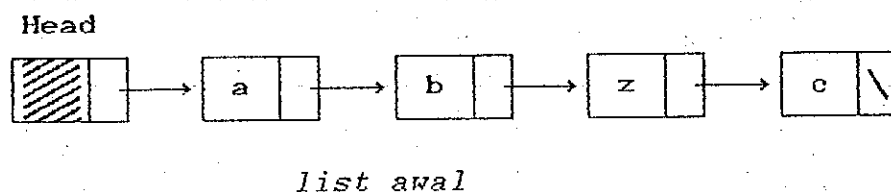
Pada list ("a" - "b" - "d") akan disisipi elemen list "c" pada tempat yang sesuai untuk membuat list menjadi : ("a" - "b" - "c" - "d").



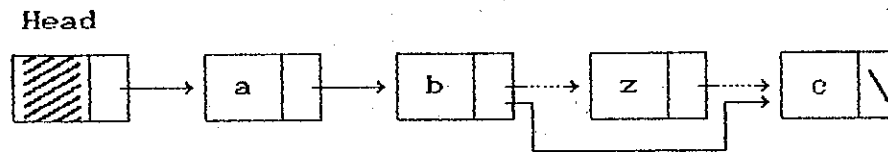
Gambar : 3.3. Penyisipan sebuah elemen

Contoh : 3.5.

Pada list ("a" - "b" - "z" - "c") akan dihapus elemen list "z" dari rangkaian tersebut, sehingga list akan menjadi : ("a" - "b" - "c").



list awal



penghapusan elemen "z"



list yang diinginkan

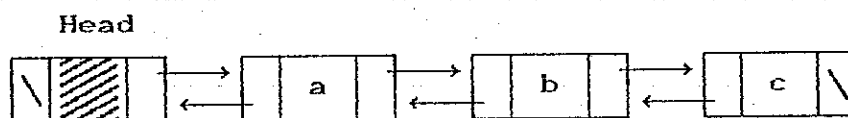
Gambar : 3.4. Penghapusan sebuah elemen

3.2.3. Tipe Data Abstrak (TDA) Linked - List

Berdasarkan beberapa prosedur - prosedur dasar yang dapat dioperasikan pada Tipe Data Abstrak Linked - List, maka dapat dimodifikasikan beberapa Tipe Data abstrak yang lain. Beberapa Tipe Data Abstrak tersebut adalah :

Doubly Linked - List

Doubly Linked - List adalah linked - list biasa, hanya saja setiap elemen list-nya memuat 2 macam pointer yang menunjuk ke elemen list sebelumnya dan menunjuk ke elemen list sesudahnya.



Gambar : 3.5. Doubly Linked - List dari

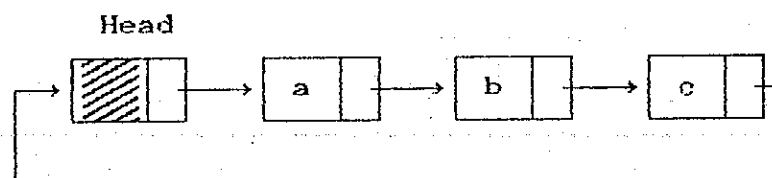
list ("a" - "b" - "c")s agree that UNDIP-IR may, without

Circular Linked - List

Pada Linked - list yang telah dibahas di atas, seluruh list diawali dengan elemen kepala list atau head dan diakhiri pada elemen list terakhir atau ujung list. Circular Linked - List berbeda dari linked - list yang lain, karena Circular Linked - List mengaitkan elemen list terakhir dengan elemen list awal.

Circular Linked - List memberi kemudahan untuk mencari elemen list tertentu, yaitu dapat dimulai dari mana saja. Jadi tidak seperti linked - list yang lain, yang harus dimulai dari elemen kepala list atau head. Karenanya pengertian elemen kepala list atau head pada Circular Linked - List tidak perlu diperhatikan.

Deklarasi struktur data dari Circular Linked - List hampir sama dengan linked - list yang lain, hanya perbedaannya terletak pada adanya pointer pengait antara ujung list dengan awal list.



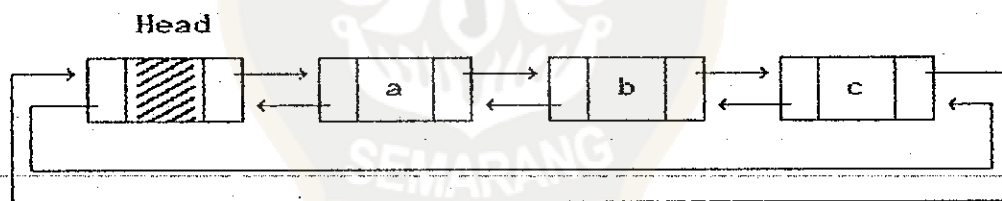
Gambar : 3.6. Circular Linked - List dari list

("a" - "b" - "c")

Doubly Linked Circular - List

Dilihat dari namanya dapat diketahui bahwa Doubly Linked Circular - List merupakan gabungan dari Doubly Linked - List dan Circular Linked - List.

Deklarasi struktur data dari Doubly Linked Circular - List juga merupakan perpaduan dari Doubly Linked - List dan Circular Linked - List, dimana setiap elemen list - nya memuat 2 macam pointer yang menunjuk pada elemen list sebelumnya dan elemen list sesudahnya, selain itu juga terdapat pointer pengait antara ujung list dan awal list.



Gambar : 3.7. Doubly Linked Circular - List dari list
("a" - "b" - "c")

Multilevel Linked - List

Telah dikatakan di atas bahwa struktur suatu list adalah cukup fleksibel, tetapi dari berbagai macam data yang ada tidak semua data dapat disimpan dalam suatu struktur list.

Contoh : 3.8.

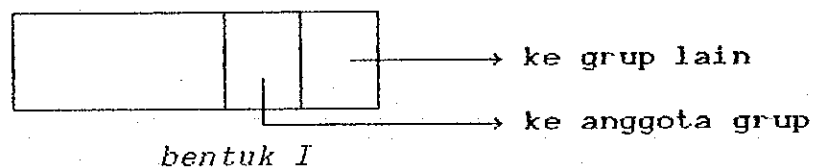
Misal data dengan model sebagai berikut :

Daftar jaga (shift)	Pagi	Sore	Malam
Petugas	A B C	D E F	G H I

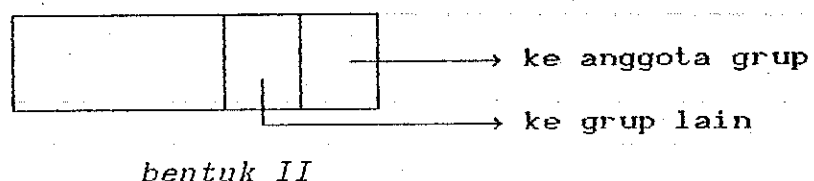
Bentuk - bentuk linked - list yang telah dijelaskan pada sub bab - sub bab di atas tidak efisien untuk menyimpan data yang memiliki bentuk demikian.

Untuk mengatasi hal tersebut, maka mula - mula dibentuk sebuah list yang memiliki elemen grup jaga (shift) : Pagi, Sore, Malam, kemudian ditambahkan sebuah pointer lagi pada elemen list tersebut yaitu pointer keanggotaan yang menunjukkan elemen - elemen anggota.

Dengan demikian bentuk list yang dimaksud adalah :



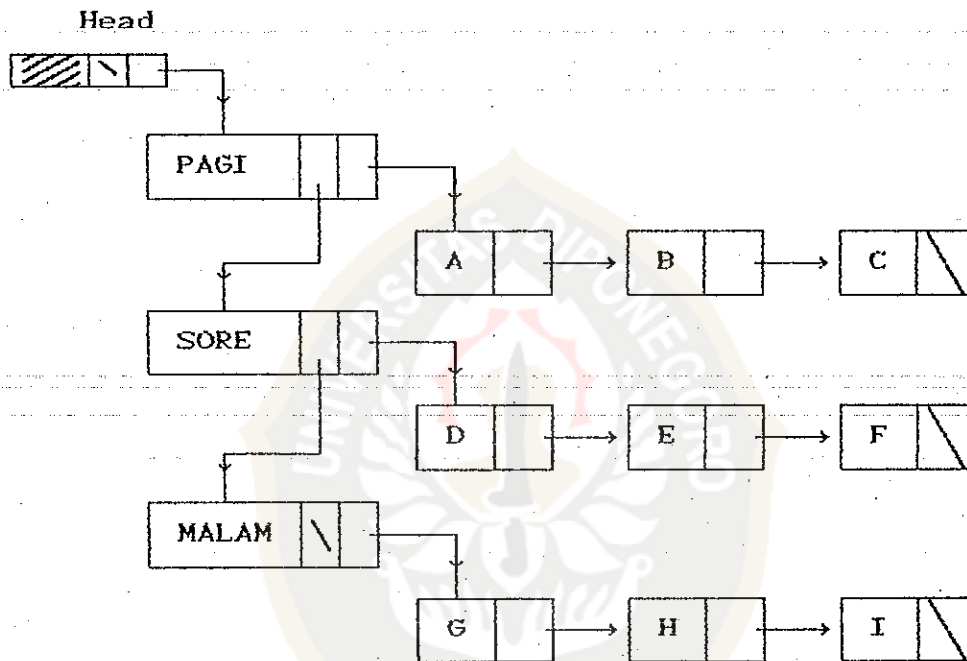
atau dapat juga dibuat dengan bentuk :



Bentuk list demikian dikenal dengan nama *Multilevel Linked - List*.

Sehingga data di atas memiliki struktur data sebagai berikut :

(dengan menggunakan bentuk II)



3.2.4. List

Elemen dari suatu struktur list dapat merupakan suatu struktur tertentu atau merupakan salah satu dari Tipe Data Abstrak yang telah dibicarakan di atas. Lebih dari itu dapat juga terjadi bahwa elemen dari suatu list juga merupakan suatu struktur list sehingga dapat dikatakan bahwa suatu struktur list yang utama mempunyai elemen berupa suatu *sub list*. Dengan demikian dapat disimpulkan bahwa elemen suatu list dapat juga merupakan list juga yang strukturnya sama dengan

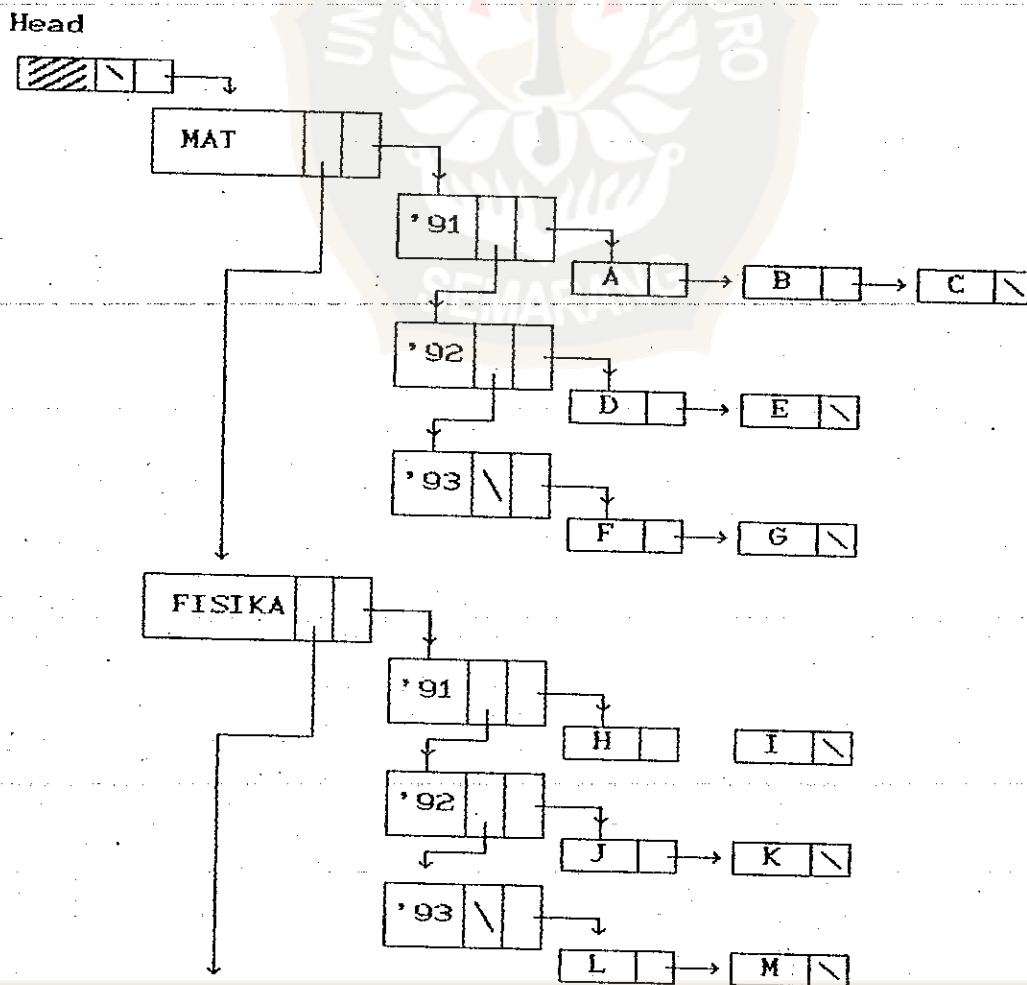
list utama. Struktur list demikian bersifat rekursif dan dikenal dengan nama *recursive list*.

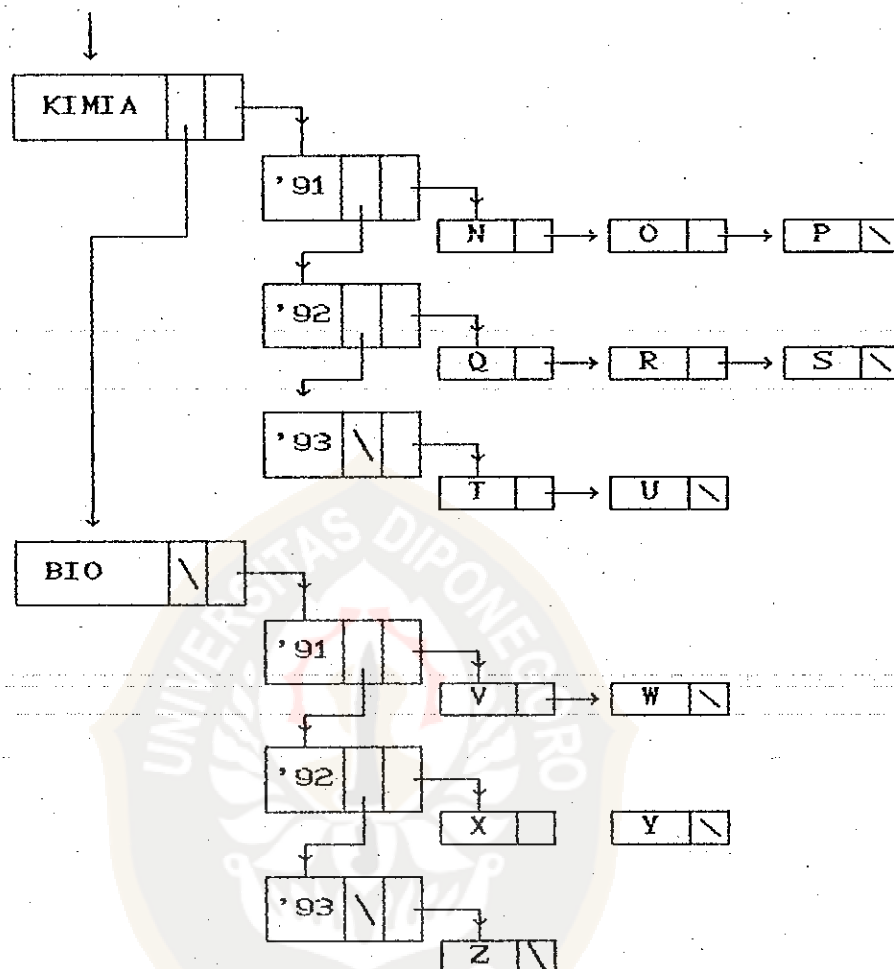
Contoh : 3.7.

Daftar Mahasiswa yang duduk sebagai Panitia OPSPEK bagi Mahasiswa Baru di Fakultas MIPA Universitas Diponegoro Semarang tahun 1994.

Jurusan	Matematika			Fisika			Kimia			Biologi		
Angkatan	91	92	93	91	92	93	91	92	93	91	92	93
Nama Mahasiswa	A B C	D E	F G	H I	J K M	L N O P	Q R S	T U	V W	X Y	Z	

Struktur data dari data tersebut adalah sebagai berikut :

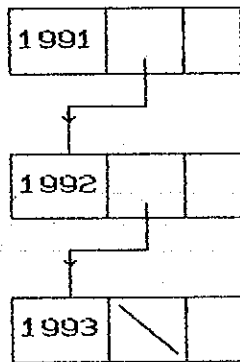




Dari contoh di atas list Jurusan (Matematika, Fisika, Kimia, Biologi) yang masing - masing memiliki elemen list Angkatan (1991, 1992, 1993) memiliki bentuk struktur list yang sama dengan elemen list - nya, yaitu Multilevel Linked - List. Struktur list dengan bentuk demikianlah yang dikenal dengan nama recursive list.

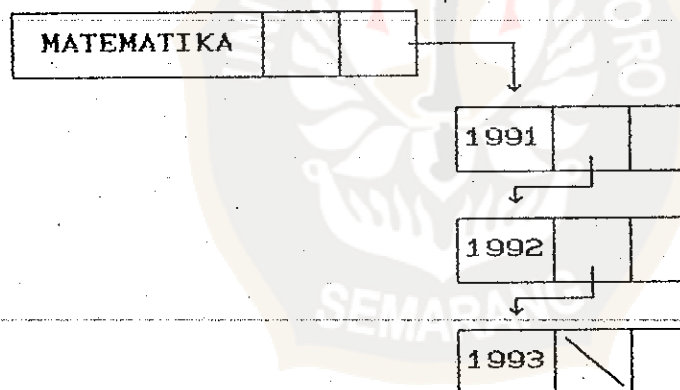
Dapat diketahui pula dari contoh di atas bahwa ada 2 macam list pada recursive list tersebut, yaitu *list alternatif* dan *list anggota*.

* List Angkatan (1991, 1992, 1993) adalah *list alternatif*.



Gambar : 3.9. List alternatif

* dan List Angkatan ((1991, 1992, 1993)) adalah *list anggota*.



Gambar : 3.10. List anggota

Dapat diketahui pula bahwa list Angkatan ((1991,1992,1993)) yang merupakan list anggota memiliki bentuk struktur yang berbeda dengan list Nama Mahasiswa (A, B, C) yang juga merupakan list anggota. Pada list Angkatan (1991, 1992, 1993) memiliki bentuk Multilevel Linked - List, tetapi pada list Nama Mahasiswa (A,B,C) memiliki bentuk linked - list biasa (Tipe Data Dasar).