

AIK21331

STRUKTUR DATA

03a ADT Stack

Departemen Ilmu Komputer/Informatika
Universitas Diponegoro

Penyegaran

- 1) Apa hubungan konsep:
selektor, mutator, inisiator ?
- 2) Apa hubungan konsep:
selektor, predikat, komponen?
- 3) Apa hubungan konsep:
komponen, elemen, objek?
- 4) Bagaimana cara mengakses elemen ADT
Tabel?
- 5) Bagaimana cara mengakses elemen ADT
Matriks?

Berpikir Komputasional (*Computational Thinking*)

- **Pilah** bagian-bagiannya (**dekomposisi**)
- **Pilih** yang penting (**abstraksi**)
- Padukan **langkah** penyelesaian (**algoritma**)
- Kenali **pola** (**pattern recognition**)

Operasi Primitif Koleksi

- Terhadap **koleksi** (unit objek)
 - 1) Penciptaan koleksi kosong (*create*)
 - 2) Penghapusan koleksi utuh (*drop*)
 - 3) Penambahan 1 elemen ke koleksi (*add/insert*)
 - 4) Penghapusan 1 elemen dari koleksi (*delete*)
- Terhadap **elemen**
 - 1) Informasi nilai elemen (*selector/get*)
 - 2) Pengubahan nilai elemen (*mutator/modifier/set*)
 - 3) Iterasi semua elemen (*traversal*)
 - 4) Pencarian nilai elemen tertentu (*search*)
 - 5) Pengurutan elemen dengan aturan khusus (*sort*)

Koleksi Objek Generik

- Pada konteks prosedural, mendefinisikan koleksi objek dilakukan dalam 3 tingkatan abstraksi :
 - 1) Definisi **Fungsional** (abstrak)
deklarasi struktur dan operator
 - 2) Representasi **Logik** (abstrak)
spesifikasi type dari struktur, belum bergantung pada memori/mesin, operator diubah menjadi fungsi/prosedur, ada parameter formal, primitif prosedural (*traversal, search*)
 - 3) Representasi/implementasi **Fisik** (konkrit)
spesifikasi dari struktur data sesuai dengan implementasinya dalam memori komputer dan kesediaan dalam bahasa pemrograman

•

Implementasi Fisik **Kontigu**

- sekumpulan data ditempatkan di memori benar-benar secara fisik adalah kontigu, setiap elemen ditaruh secara berturutan posisi alamatnya dengan elemen lain.
- untuk mencapai elemen berikutnya, cukup melalui suksesor alamat yang sedang "current".
- Agar suksesor tetap terdefinisi, alokasi struktur ini sudah ditetapkan sebelumnya, supaya data tidak melompat.
- Disebut **statis** karena alokasi memori dilakukan sekaligus untuk seluruh koleksi.
- Dibatasi kapasitas yang dideklarasikan di awal

Implementasi Fisik **Berkait**

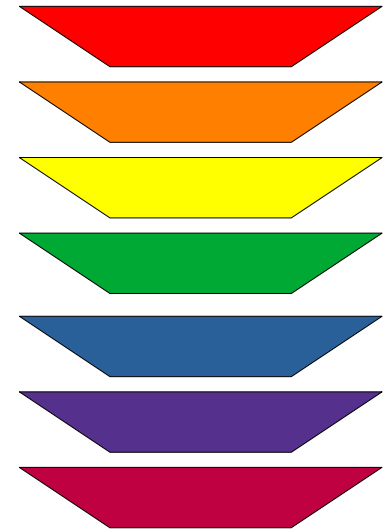
- sekumpulan data ditempatkan di memori komputer dapat terpencar, namun dapat ditelusuri karena ada informasi berupa alamat (*address*), yang menghubungkan elemen yang satu dengan yang lain.
- Karena alamat untuk mencapai elemen lain ada secara eksplisit, alamat yang akan dipakai dapat saja dialokasikan pada waktunya ataupun sudah ditetapkan dari awal.
- Disebut juga struktur **dinamis** karena ruang memori yang digunakan dapat membesar dan mengecil sesuai kebutuhan.
- Dibatasi kapasitas memori fisik komputer

Tipe Komposit Majemuk

- ADT Tabel : komponen wadah 1 dimensi vs ukuran, posisi elemen tunggal
- ADT Matriks : komponen wadah N dimensi vs ukuran, posisi elemen tuple <baris, kolom>
- ADT **Stack** : akses elemen 1 pintu, elemen kunci di puncak/top
- ADT Queue : akses elemen 2 pintu, elemen kunci di depan/kepala dan di belakang/ekor
- ADT String : elemen-elemen dapat bergabung menjadi himpunan bagian
- ADT List : elemen saling terkait setara
- ADT Tree : elemen berkait secara hierarkis

Koleksi Tumpukan

- Penambahan dan pengurangan elemen lewat 1 pintu, elemen kunci di puncak/top
- Contoh: tumpukan peti kemas, tumpukan piring
- Operasi: tumpuk/tindih (*push*) , angkat (*pop*)



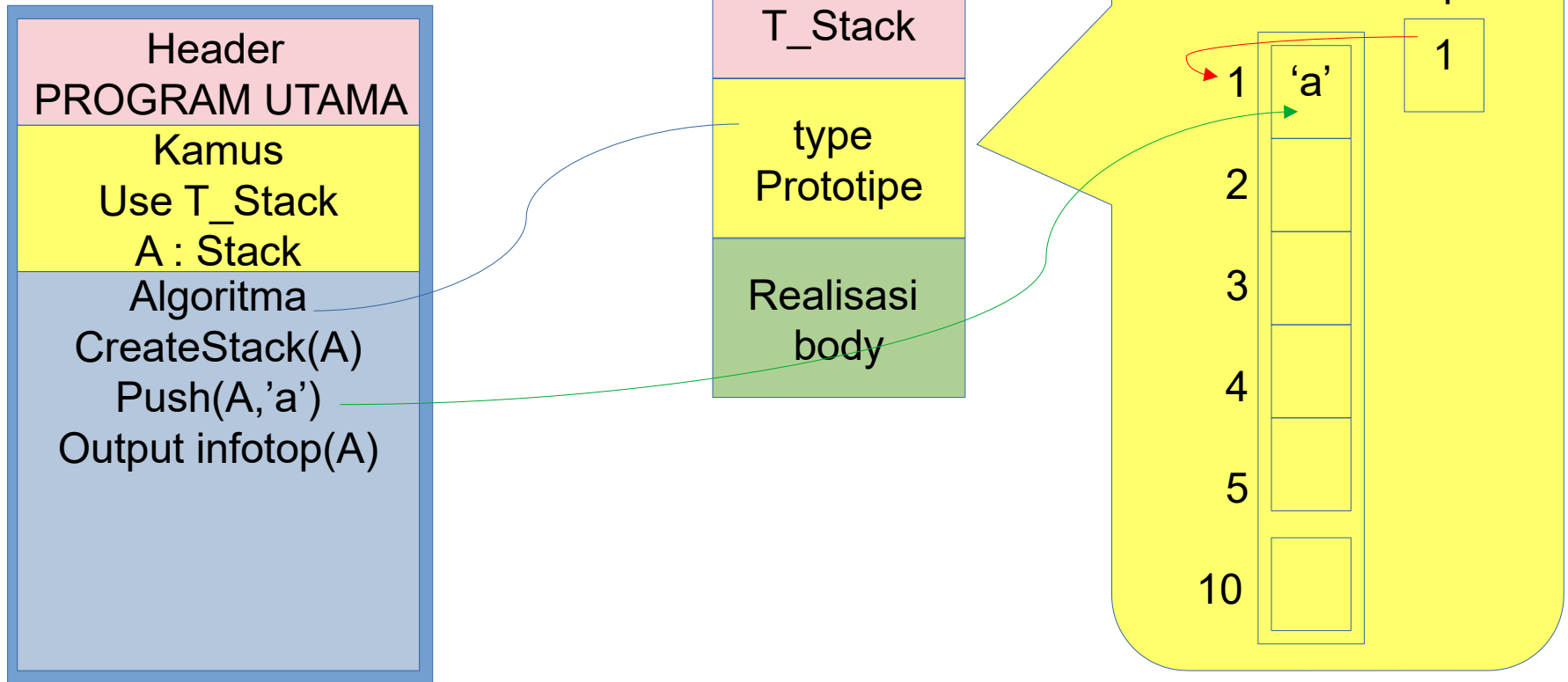
ADT Stack

- Objek unit kolektif 1 dimensi
- Memiliki elemen anggota, mungkin kosong
- Setiap elemen memiliki nilai (*value/info*)
- Elemen terpenting **TOP** / puncak
- Akses elemen 1 pintu, yaitu di puncak
- *Last In First Out (LIFO)*
- Contoh penerapan :
 - pemanggilan prosedur/fungsi: $f(g(x))$,
 - eksekusi ekspresi aritmetika: $2+5*7$,
 - rekursivitas: $f(x) = x + f(x-1)$ & $f(0)=1$,
 - backtracking: tombol undo di editor menyimpan status perubahan

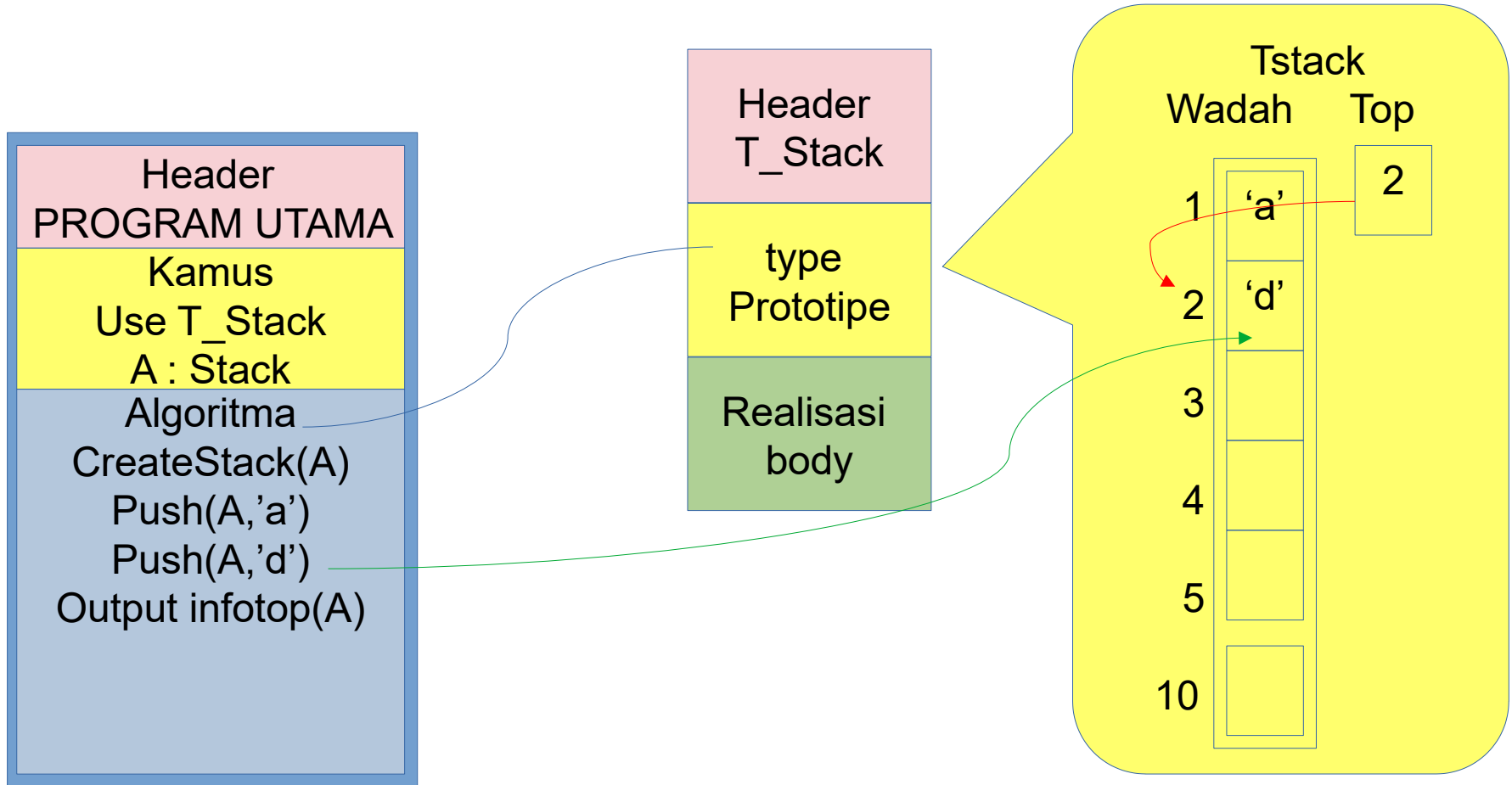
Definisi Fungsional Stack

- **CreateStack** : $\rightarrow S$
{ Membuat stack kosong }
- **IsEmptyStack** : $S \rightarrow \text{boolean}$
{ Test stack kosong }
- **IsFullStack** : $S \rightarrow \text{boolean}$
{ Test stack penuh }
- **Push** : $e \times S \rightarrow S$
{ Menambahkan sebuah elemen e sebagai TOP, TOP berubah nilainya }
- **Pop** : $S \rightarrow S \times e$
{ Mengambil nilai elemen TOP, sehingga TOP baru adalah elemen yang masuk sebelum elemen TOP, mungkin S menjadi kosong }
- Selektor : $\text{Top}(S)$ mengembalikan posisi top
- Selektor : $\text{Infotop}(S)$ mengembalikan nilai top

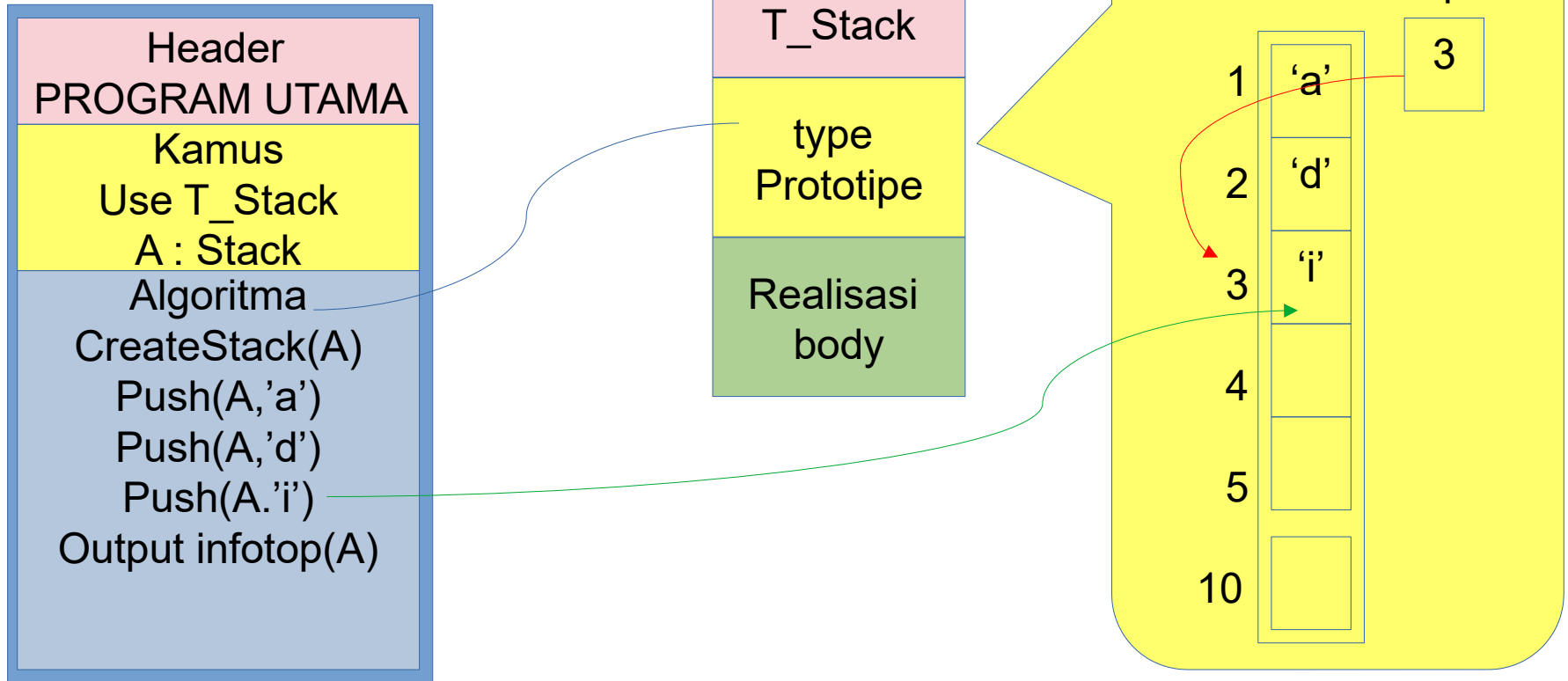
ADT Stack **PUSH**



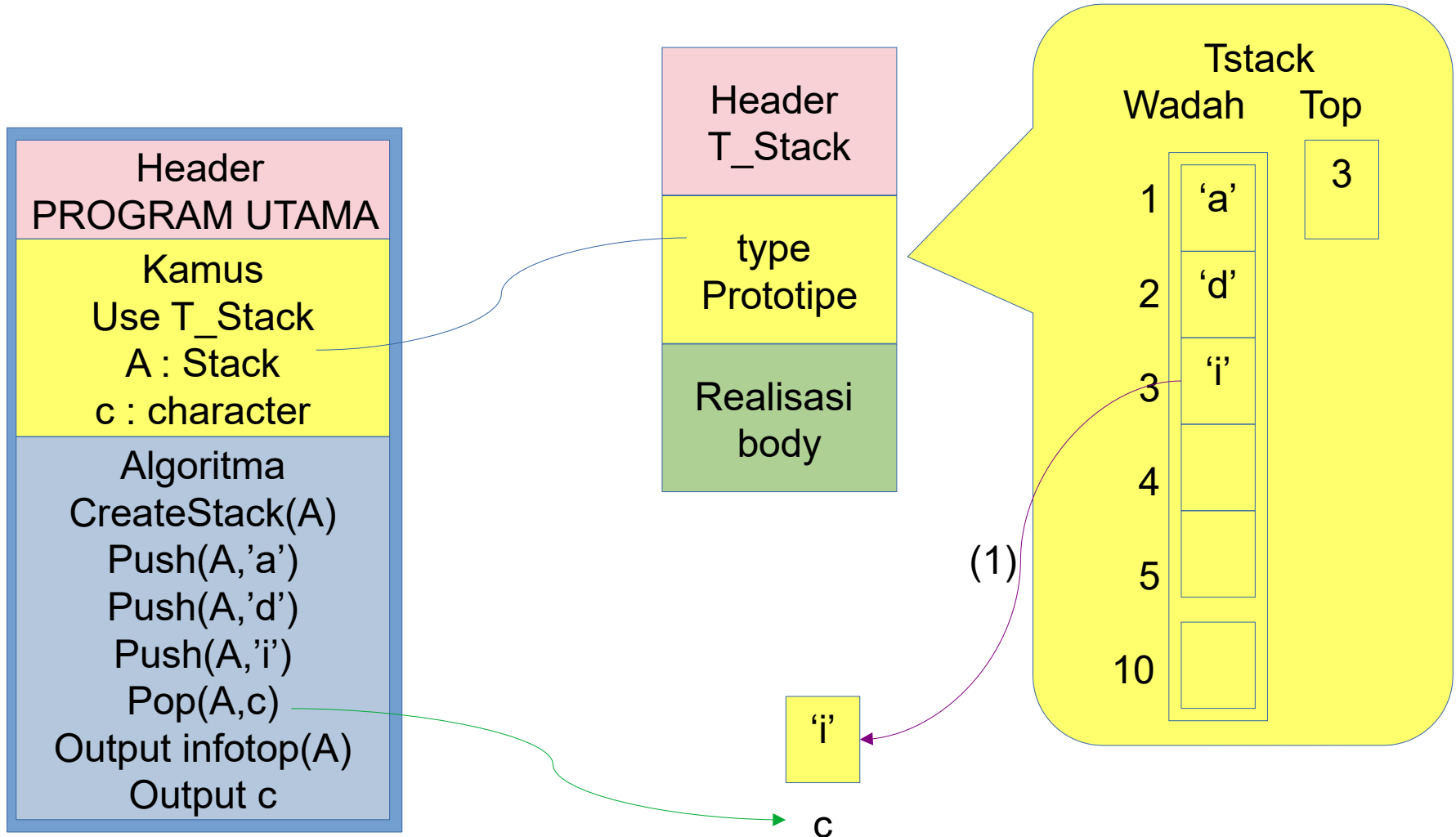
ADT Stack **PUSH**



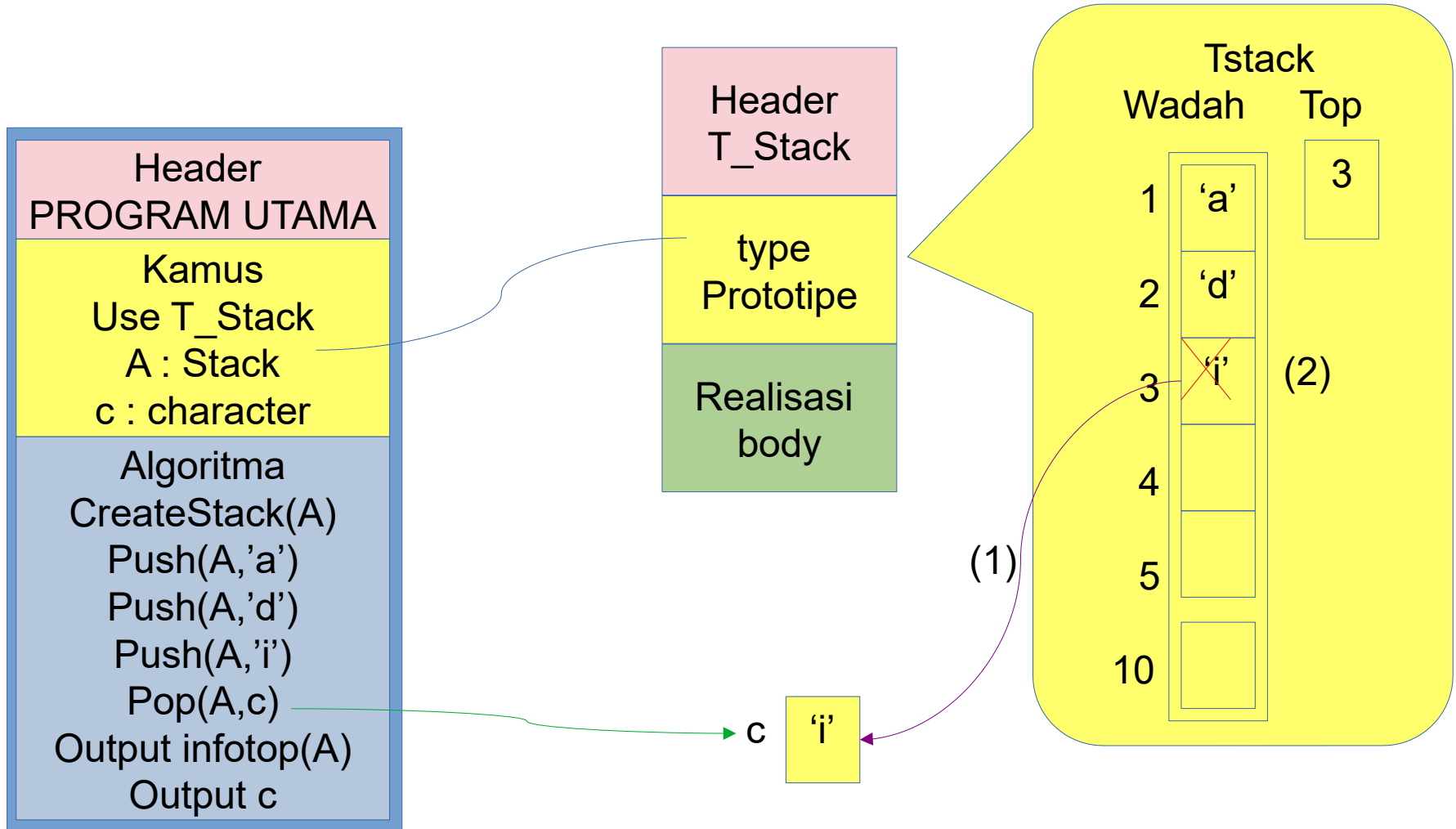
ADT Stack **PUSH**



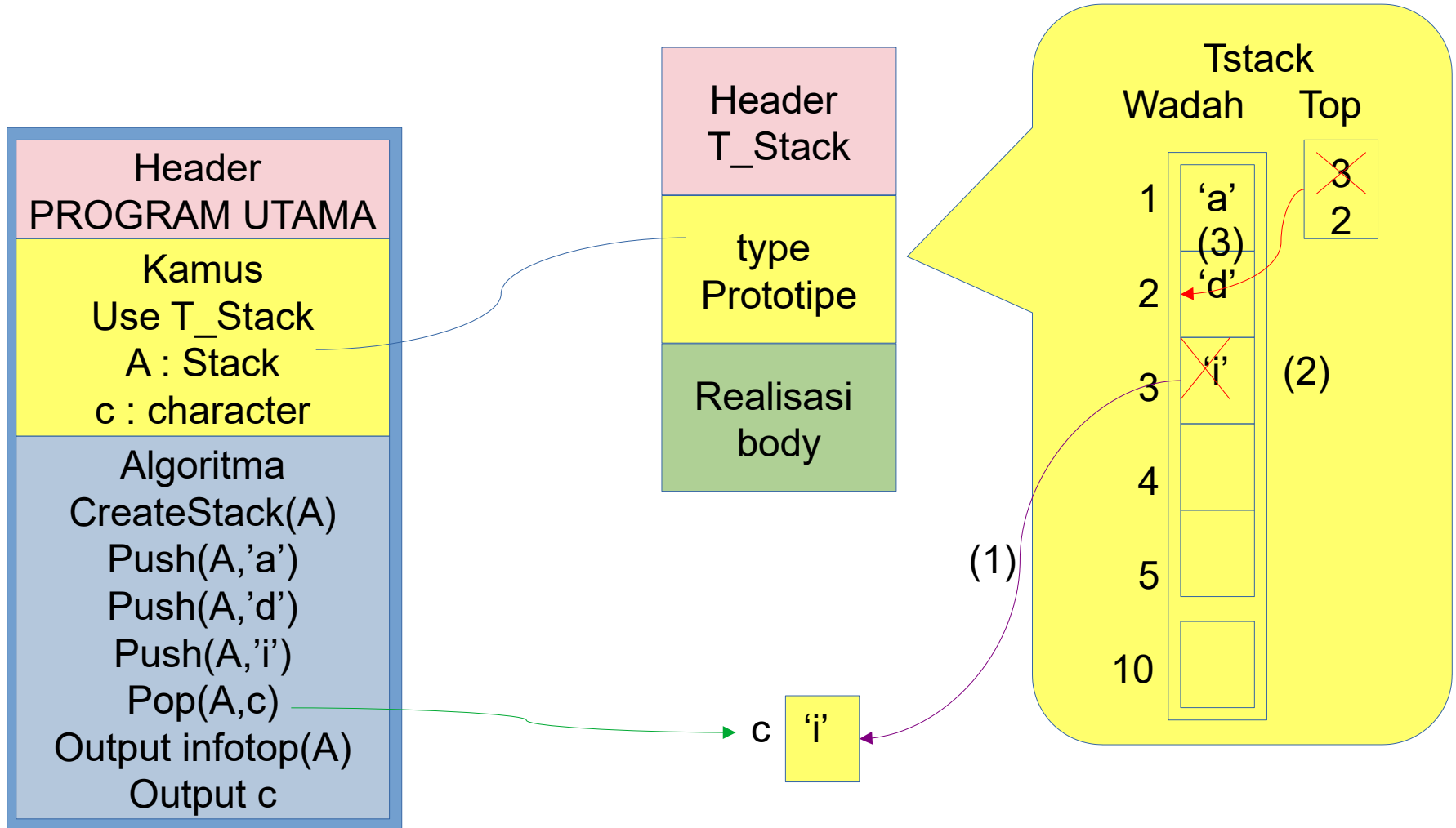
ADT Stack POP



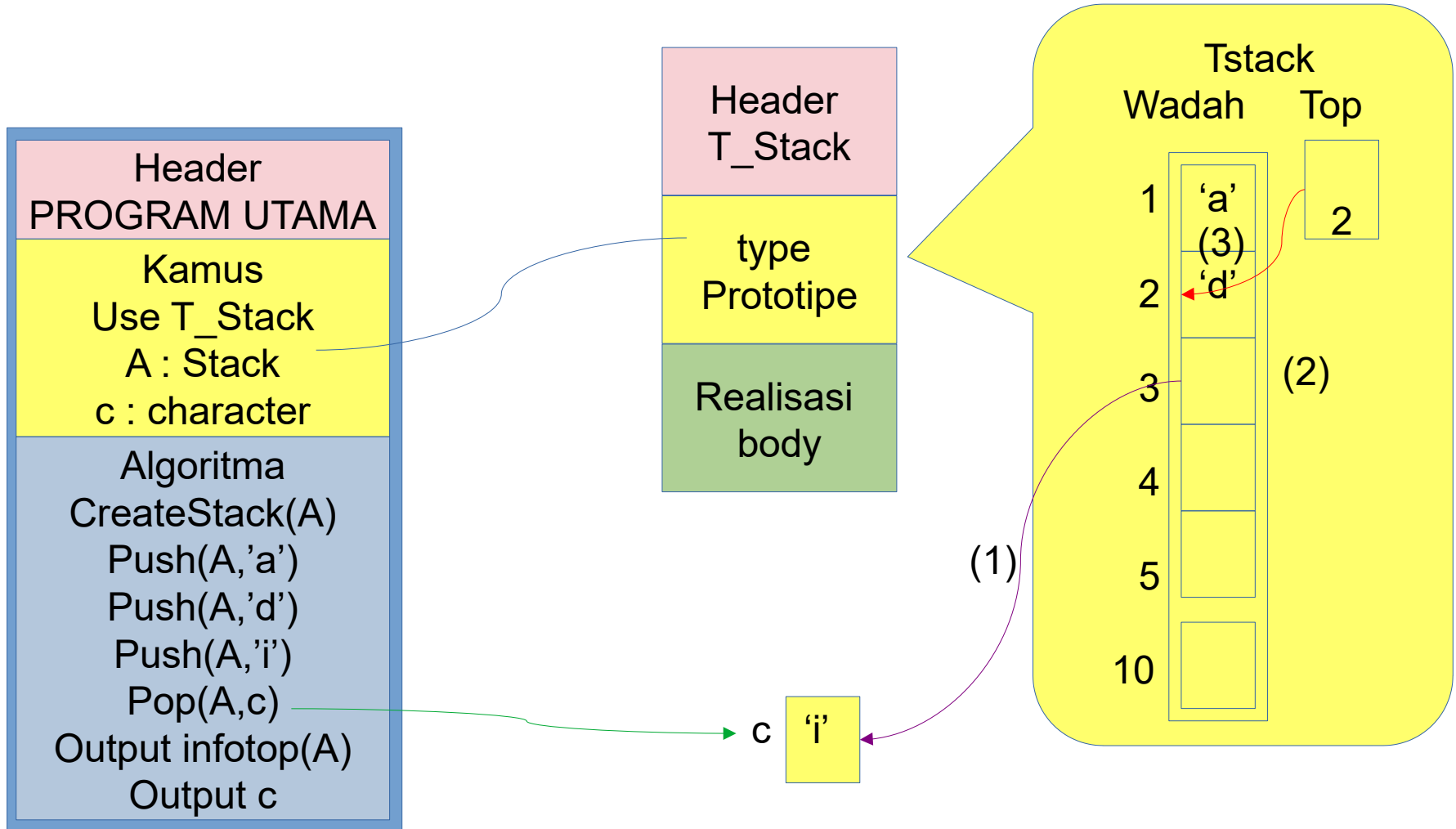
ADT Stack POP



ADT Stack POP



ADT Stack POP



Type Stack (kontigu)

Type TStack = < wadah : array[1..10] of character ,
top : integer >

- Primitif:
 - Initiator: CreateStack
 - Selektor: Top(S), Infotop(S) = S.wadah[S.top]
 - Mutator: push, pop
 - Predikat: isEmptyStack, isFullStack
 - Operator: printStack

Modul T_Stack

DEKLARASI/DEFINISI&SPESIFIKASI TIPE & PROTOTYPE

Type Tstack = <wadah:array[1..10] of character, top:integer>

Procedure **CreateStack**(output S:Tstack)

{I.S: - ; F.S: S terdefinisi }

{Proses mengisi elemen wadah dengan ' ', top 0}

Function **Infotop**(S:Tstack) -> character

{mengembalikan nilai elemen puncak}

Function **Top**(S:Tstack) -> integer

{mengembalikan posisi puncak}

Function **isEmptyStack**(S:Tstack) -> boolean

{mengembalikan true bila S kosong}

Function **isFullStack**(S:Tstack) -> boolean

{mengembalikan true bila S penuh}

Procedure **Push**(input/output S:Tstack, input e:character)

{I.S: S,e terdefinisi, S mungkin kosong }

{F.S: S tetap, atau infotop(S)=e }

{Proses mengisi elemen e ke puncak S, bila belum penuh}

Procedure **Pop**(input/output S:Tstack, output e:character)

{I.S: S terdefinisi, mungkin kosong }

{F.S: S tetap, atau e berisi infotop(S) lama }

{Proses menghapus elemen e dari puncak S, bila belum kosong}

Procedure **PrintStack**(input S:Tstack)

{I.S:-; F.S:-; Proses: menampilkan info elemen S }

Sumber Pembelajaran

- Inggriani Liem. Diktat Struktur Data. 2008
- Niclaus Wirth. **Algorithms and Data Structures.** 2004
- Standish, Thomas A. Data Structures, Algorithms, & Software Principles in C. Addison Wesley Publishing Company 1995
- AHO, Alfred V., John E. Hopcroft, Jeffrey D. Ullman. Data Structures and Algorithm. Addison Weshley Publishing Compani.1987