

AIK21331

STRUKTUR DATA

04a ADT Queue

Departemen Ilmu Komputer/Informatika
Universitas Diponegoro

Penyegaran

- 1) Apa hubungan konsep:
konstruktor, iterasi, inisialisasi ?
- 2) Apa hubungan konsep:
isFullStack, isEmptyStack, komponen ?
- 3) Bagaimana cara menambah/mengurangi
elemen ADT Stack ?
- 4) Apa hubungan konsep:
Push, Pop, variabel ?
- 5) Bagaimana cara mengakses struktur
sebuah **variabel** S bertipe TStack?

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

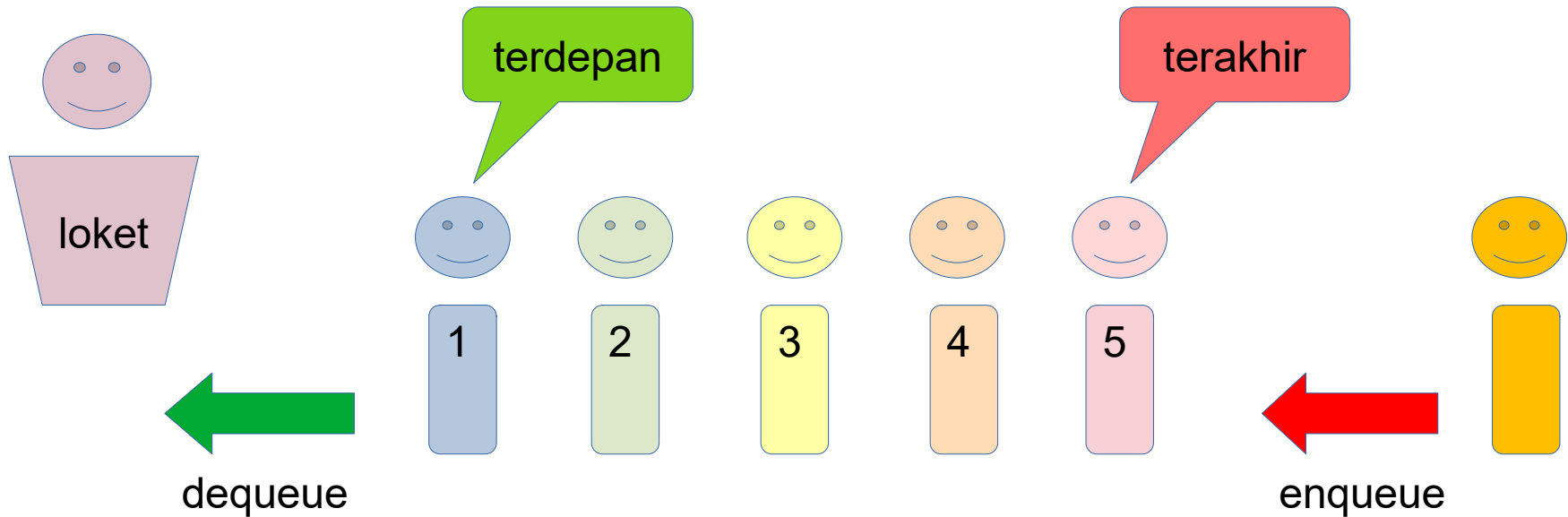
- Pada konteks prosedural, mendefinisikan koleksi objek dilakukan dalam 3 tingkatan abstraksi :
 - 1) Definisi **Fungsional** (abstrak)
deklarasi struktur dan operator
 - 2) Representasi **Logik** (abstrak)
spesifikasi type dari struktur, belum bergantung pada memori/mesin, operator diubah menjadi fungsi/prosedur, 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
Macam: **kontigu** vs **berkait**

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 Antrian

- Penambahan elemen di belakang(ekor) sedangkan pengurangan elemen di depan (kepala)
- Contoh: antrian di kasir, antrian pasien
- Operasi: antre (*enqueue*), dilayani (*dequeue*)



ADT Queue

- Objek unit kolektif 1 dimensi
- Memiliki elemen anggota, mungkin kosong
- Setiap elemen memiliki nilai (*value/info*)
- Elemen terpenting **HEAD** dan **TAIL**
- Akses elemen 2 pintu, yaitu di depan, di belakang
- *First In First Out (FIFO)*
- Contoh penerapan :
 - eksekusi baris assignment dalam algoritma
 - $f(x)$; $g(x)$;
 - penghapusan status di tombol undo di editor
 - menyimpan status perubahan

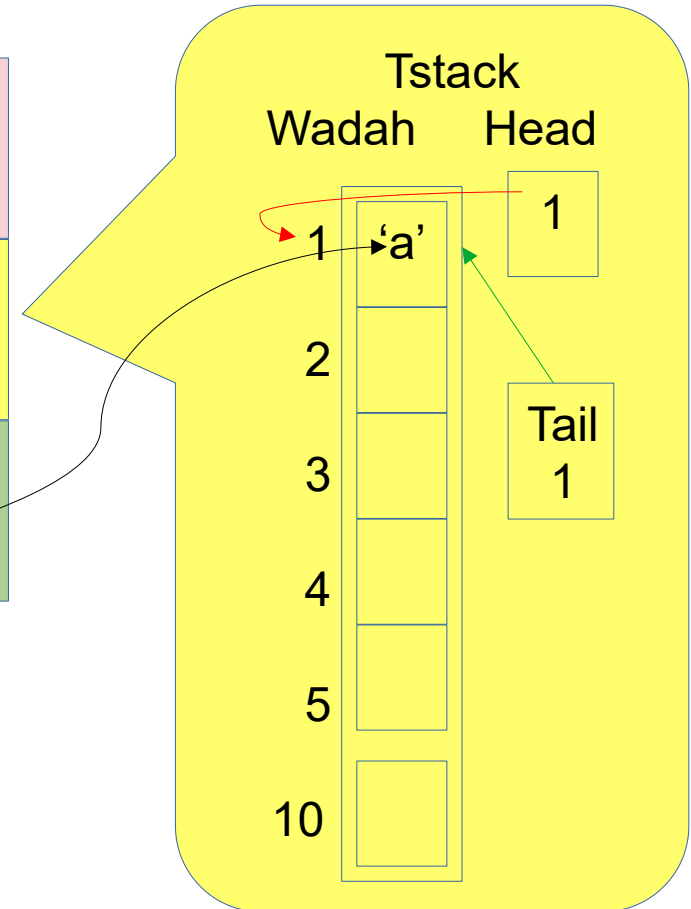
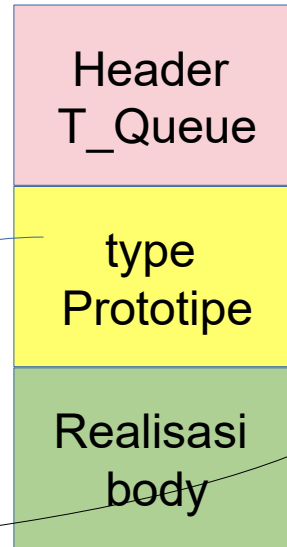
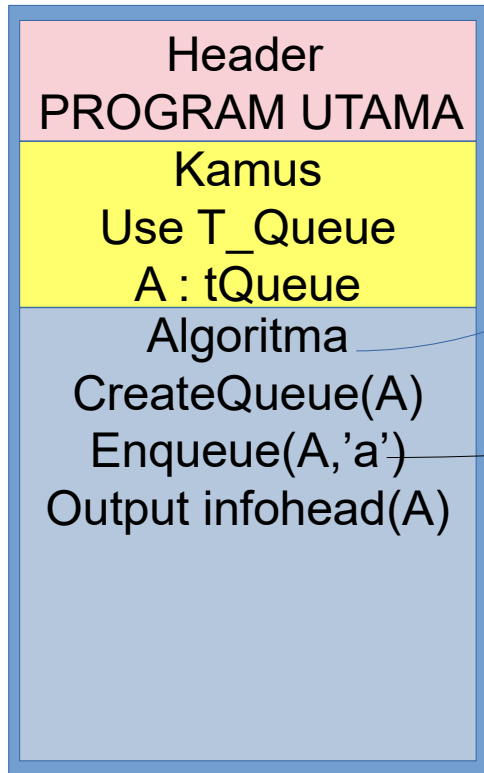
Definisi Fungsional Stack

- **CreateQueue** : $\rightarrow Q$
{ Membuat Queue kosong }
- **IsEmptyQueue** : $Q \rightarrow \text{boolean}$
{ Test Queue kosong }
- **IsFullQueue** : $Q \rightarrow \text{boolean}$
{ Test Queue penuh }
- **Enqueue** : $e \times Q \rightarrow Q$
{ Menambahkan elemen e sebagai Tail Q }
- **Dequeue** : $Q \rightarrow Q \times e$
{ Mengambil nilai elemen Head, sehingga head baru adalah elemen yang masuk setelah elemen Head, mungkin Q menjadi kosong }
- Selektor : Head(S) mengembalikan posisi kepala;
Tail(S) mengembalikan posisi ekor;
InfoHead(S) mengembalikan nilai kepala;
InfoTail(S) mengembalikan nilai ekor;

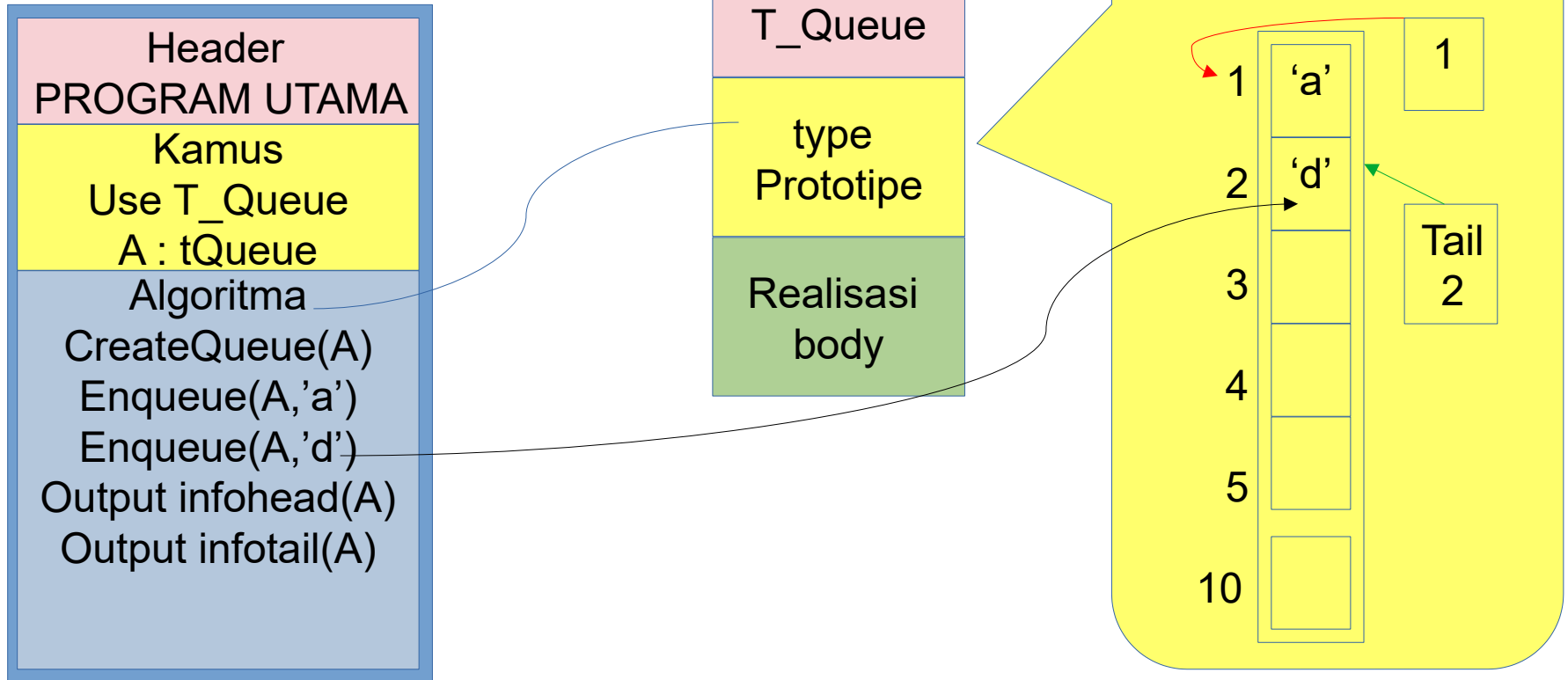
ADT Queue

- Beberapa model struktur logik, perhatikan soal!
- 1. Model kontigu, head **stabil**
posisi head selalu 0 atau 1, bila terjadi dequeue maka semua elemen bergeser maju 1 langkah. Contoh: antrian di kasir swalayan.
- 2. Model kontigu, head **bergeser**
posisi head bergeser dari 0,1,2... hingga posisi tertentu, bila antrian penuh maka posisi head di-reset menjadi 1, semua elemen bergeser. Contoh: antrian Periksa, posisi duduk pasien tidak geser.
- 3. Model kontigu, head **berputar**
posisi head dan tail bergeser, terutama bila sampai di kapasitas dan belum penuh. Contoh: antrian meja makan putar.

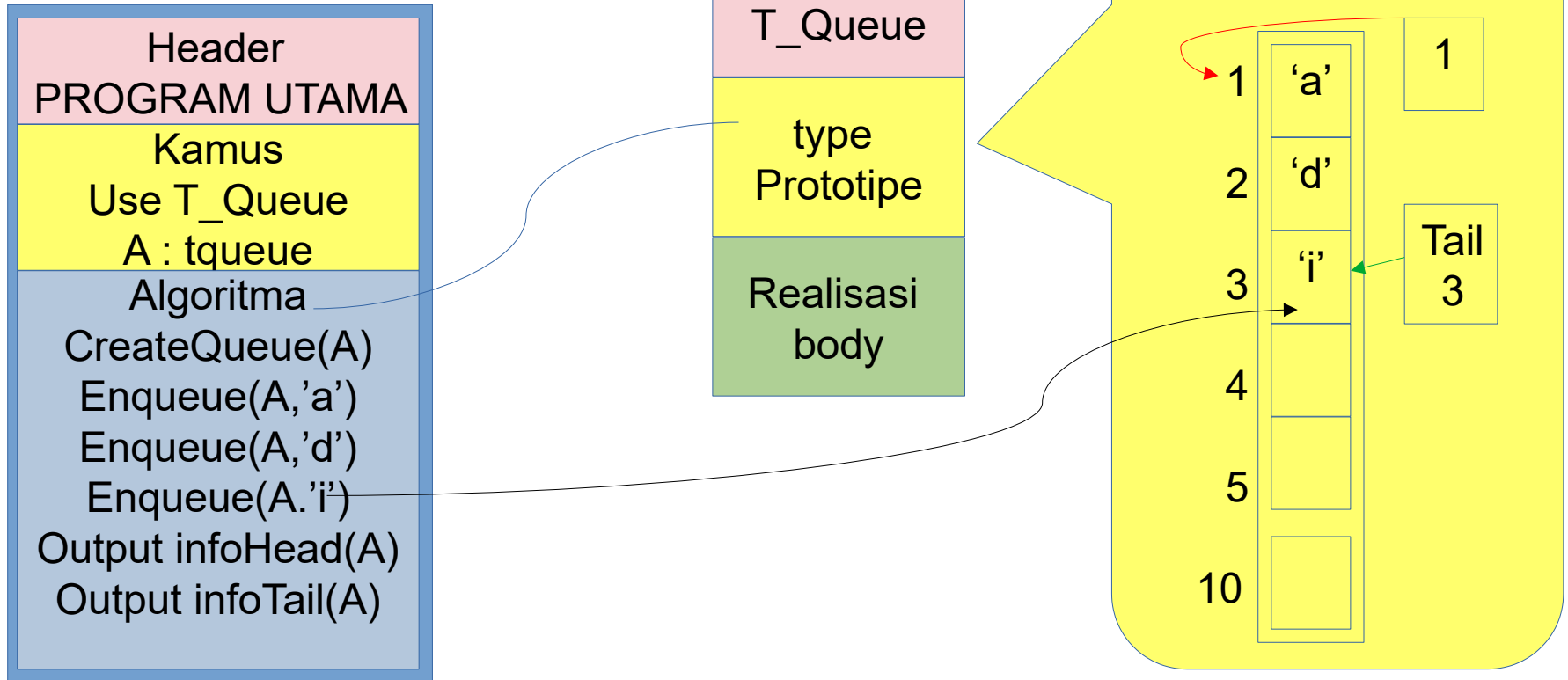
ADT Queue **ENQUEUE**



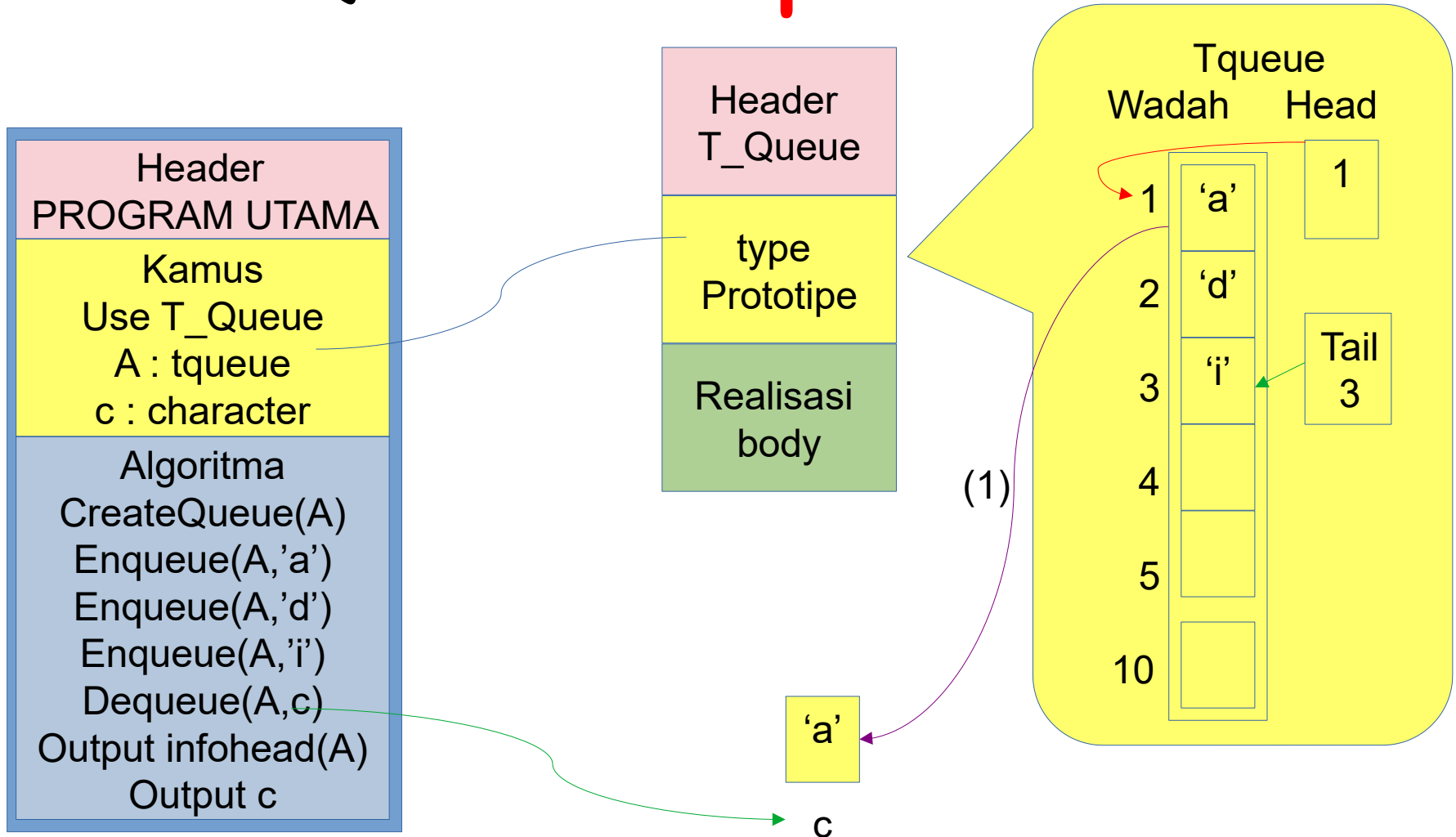
ADT Queue **ENQUEUE**



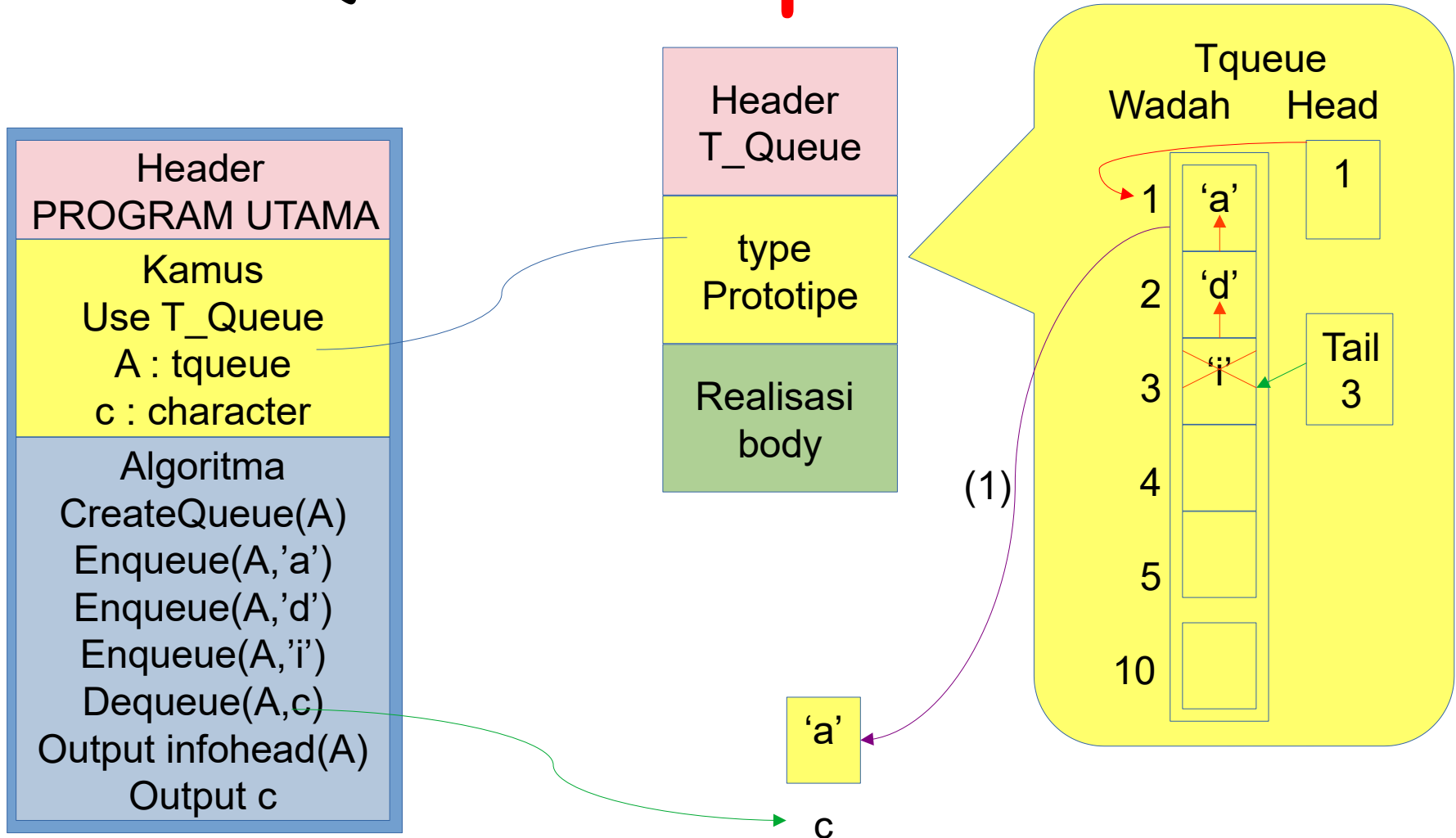
ADT Queue **ENQUEUE**



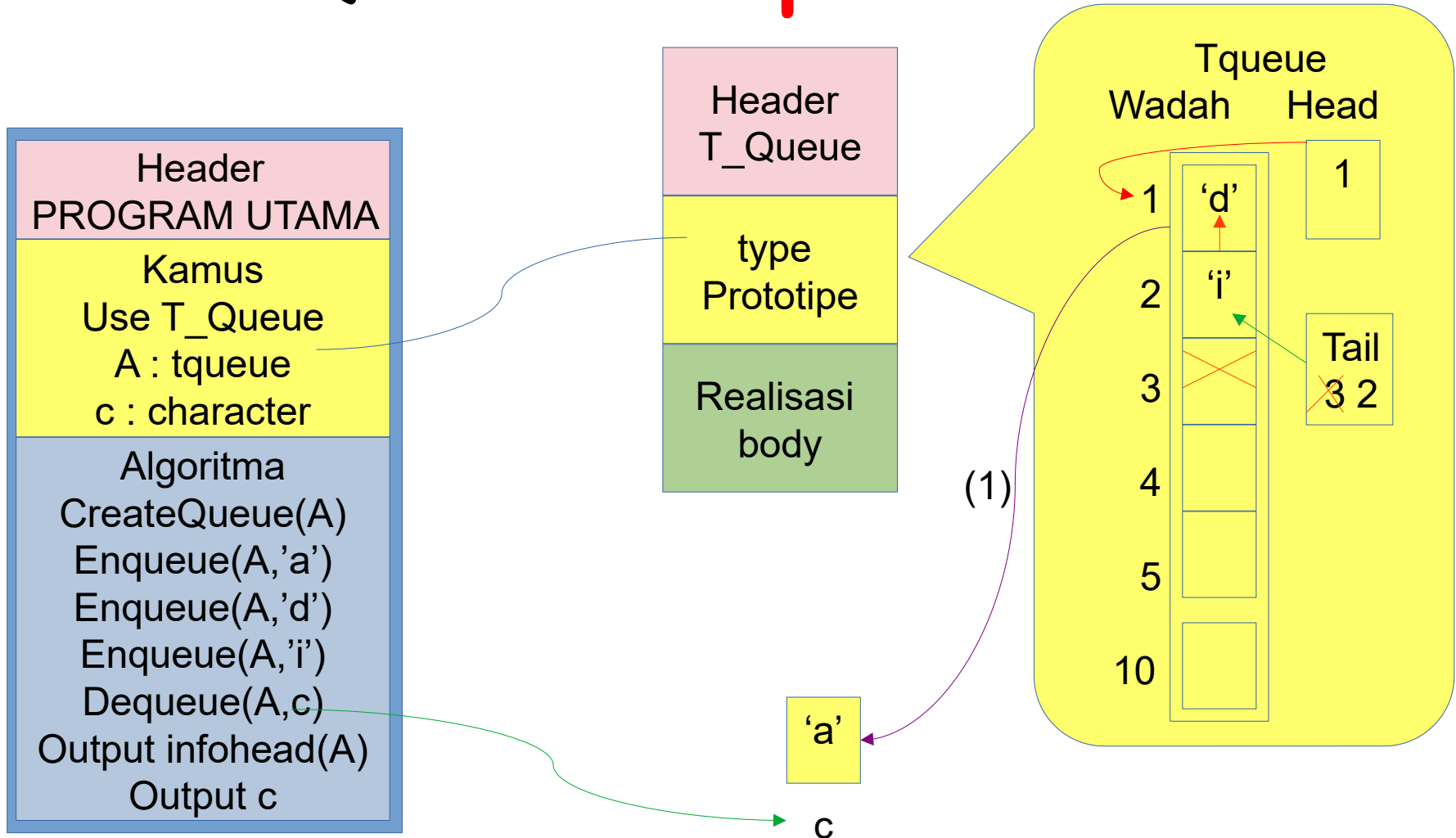
ADT Queue Dequeue



ADT Queue Dequeue



ADT Queue Dequeue



Tipe Queue (kontigu)

```

Type TQueue = < wadah : array[1..10] of character ,
                  head : integer ;
                  tail: integer >

```

- Primitif:

- Initiator: CreateQueue
- Selektor: Head, InfoHead, Tail, InfoTail
- Mutator: enqueue, dequeue
- Predikat: isEmptyQueue, isFullQueue
- Operator: printQueue, viewQueue

Modul T_Stack

DEKLARASI/DEFINISI&SPESIFIKASI TIPE & PROTOTYPE

Type TQueue = <wadah:array[1..10] of character,
 head:integer,
 tail:integer >

{Queue model I, kondisi head 0 atau 1}

{pergeseran maju pada elemen ketika dequeue}

Procedure **CreateQueue**(output Q:TQueue)

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

{Proses: mengisi elemen wadah dengan ' ', head 0, tail 0}

Function **Head**(Q:TQueue) -> integer

{mengembalikan posisi elemen terdepan}

Function **Tail**(Q:TQueue) -> integer

{mengembalikan posisi elemen terakhir}

Function **InfoHead**(Q:TQueue) -> character

{mengembalikan nilai elemen terdepan}

Function **InfoTail**(Q:TQueue) -> character

{mengembalikan nilai elemen terakhir}

Function **isEmptyQueue**(Q:TQueue) -> boolean

{mengembalikan true bila Q kosong}

Function **isFullQueue**(Q:TQueue) -> boolean

{mengembalikan true bila Q penuh}

DEKLARASI/DEFINISI&SPESIFIKASI TIPE & PROTOTYPE

Procedure **Enqueue** (input/output Q:TQueue, input e:character)

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

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

{Proses menambah elemen e ke ekor Q bila belum penuh}

Procedure **Dequeue** (input/output Q:TQueue, output e:character)

{I.S: Q terdefinisi, mungkin kosong }

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

{Proses menghapus elemen e dari head Q bila belum kosong}

{lalu geser maju 1 langkah semua elemen di belakang head}

Procedure **PrintQueue** (input Q:TQueue)

{I.S:-; F.S:-; Proses: menampilkan kondisi wadah Q }

Procedure **ViewQueue** (input Q:TQueue)

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

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