



QUICK SORT

- Dasar pemikiran : pertukaran yang efektif adalah pertukaran dengan jarak yang jauh
- Misalnya : ada n item dengan urutan terbalik. Untuk mengurutkannya diperlukan $n/2$ pertukaran. Caranya, menukar item paling kiri dengan item paling kanan dan bergerak ke tengah
- Masalahnya, bagaimana jika kuncinya acak ?



Algoritma

1. Ambil sebuah item secara acak $\Rightarrow x$
2. Scan array dari kiri dan ditemukan item $a_i > x$
3. Scan array dari kanan dan ditemukan item $a_j < x$
4. Tukar a_i dengan a_j
5. Ulangi proses scan dan tukar sampai bertemu di tengah array





Hasil Partisi

Hasilnya array terbagi menjadi 2 bagian

- Bagian kiri, kuncinya $< x$
- Bagian kanan, kuncinya $> x$

=> X bertindak sebagai sentinel



Algoritma partisi

$I = 1 ; J = n$

Pilih x

Repeat

 while $a(I) < x$ do $I = I + 1$;

 while $x < a(J)$ do $J = J - 1$;

 if $I \leq J$ then

 begin

$w = a(I); a(I) = a(J); a(J) = w$

$I = I + 1 ; J = J - 1$

 end

Until $I > J$



INSTITUT
TEKNOLOGI
HARAPAN
BANGSA

Veritas vos liberabit



Algoritma partisi

- Algoritma partisi di atas diberlakukan pada tiap-tiap partisi baru, sampai tinggal satu item.



INSTITUT
TEKNOLOGI
HARAPAN
BANGSA
Veritas vos liberabit

Algoritma rekursif Quick Sort

```
Procedure qs(L,R : integer);
Begin
  I := L ; J := R ;
  x := a[(L+R) div 2] ;
  repeat
    while a[I] < x do I := I + 1 ;
    while x < a[J] do J := J - 1 ;
    if I <= J then
      begin
        w:=a[I]; a[I]:= a[J]; a[J]:=w ;
        I := I+1; J:=J-1 ;
      end ;
  until I > J ;
  .....
  if L<J then qs(L,J);
  if I<r then qs(I,R) ;
End ;
```

