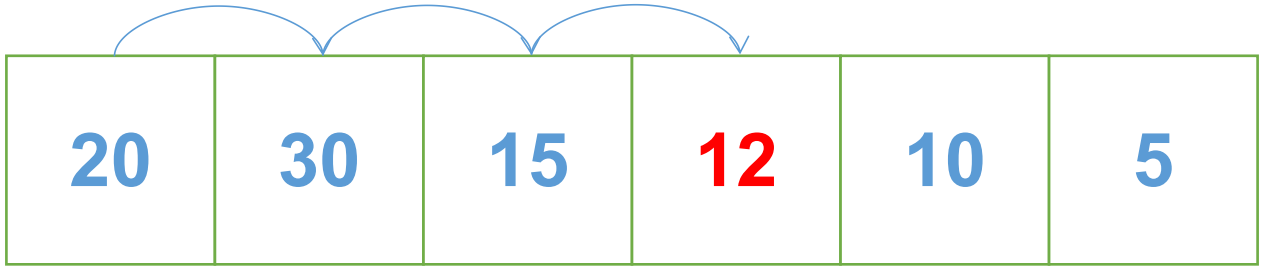
Pencarian (Searching)

Sequential Search

Data : [20, 30, 15, 12, 10, 5]

Cari : 12



Sequential Search

Data : [20, 30, 15, 12, 10, 5]

Cari : 12

12 == 20 ?

Sequential Search

Data : [20, 30, 15, 12, 10, 5]

Cari : 12

12 == 30 ?

Sequential Search

Data : [20, 30, 15, 12, 10, 5]

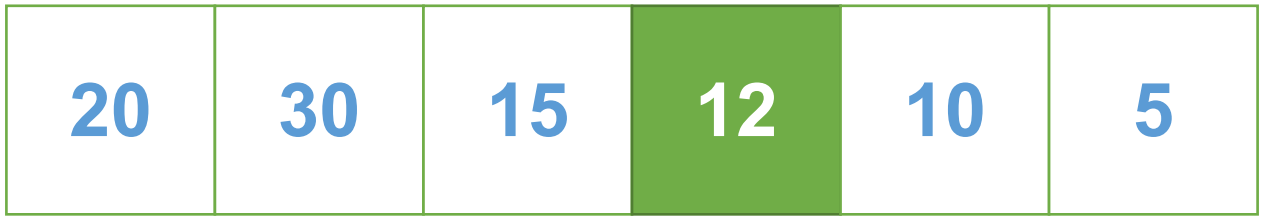
Cari : 12

12 == 15 ?

Sequential Search

Data : [20, 30, 15, 12, 10, 5]

Cari : 12

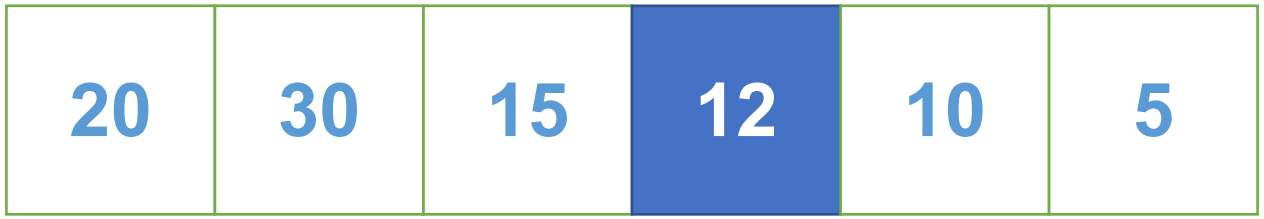


12 == 12 ?

Sequential Search

Data : [20, 30, 15, 12, 10, 5]

Cari : 12



Implementation

```
int linearSearch(int arr[], int data) {
    // Cari secara urut
    for(int i=0; i<n; i++) {
        // Ketemu, return index.
        if(arr[i] == data) {
            return i;
        }
    }
    return -1;
}
```

Binary Search

HARUS URUT

Cari : 40

20	30	40	50	70	80	90

Binary Search

Cari == data tengah, selesai

Cari < data tengah, buang kanan

Cari > data tengah, buang kiri

Cari : 40

20	30	40	50	70	80	90

Binary Search

Cari == data tengah, selesai

Cari < data tengah, buang kanan

Cari > data tengah, buang kiri

Cari : 40

20	30	40

Binary Search

Cari == data tengah, selesai

Cari < data tengah, buang kanan

Cari > data tengah, buang kiri

Cari : 40

20	30	40

Binary Search

Cari == data tengah, selesai

Cari < data tengah, buang kanan

Cari > data tengah, buang kiri

Cari : 40

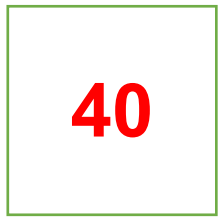
Binary Search

Cari == data tengah, selesai

Cari < data tengah, buang kanan

Cari > data tengah, buang kiri

Cari : 40



Binary Search

Cari == data tengah, selesai

Cari < data tengah, buang kanan

Cari > data tengah, buang kiri

Cari : 40

Binary Search

Data : [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

Cari : 20

Mid = (Low + High) / 2

Mid = (0 + 10) / 2 = 5

Binary Search

Data : [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

Cari : 20

20 == 60 ?

20 < 60 ?

20 > 60 ?

Binary Search

Data : [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

Cari : 20

mid = (low + high) / 2
= (0 + 4) / 2
= 2

high = mid - 1
= 5 - 1
= 4

Binary Search

Data : [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

Cari : 20

mid = (low + high) / 2
= (0 + 4) / 2
= 2

high = mid - 1
= 5 - 1
= 4

Binary Search

Data : [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

Cari : 20

20 == 30 ?

20 < 30 ?

20 > 30 ?

Binary Search

Data : [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

Cari : 20

mid = (low + high) / 2
= (0 + 1) / 2
= 0

high = mid - 1
= 2 - 1
= 1

Binary Search

Data : [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

Cari : 20

mid = (low + high) / 2
= (0 + 1) / 2
= 0

high = mid - 1
= 2 - 1
= 1

Binary Search

Data : [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

Cari : 20

20 == 10

20 < 10

20 > 10

Binary Search

Data : [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

Cari : 20

mid = (low + high) / 2
= (1 + 1) / 2
= 1

low = mid + 1
= 0 + 1
= 1

Binary Search

Data : [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

Cari : 20

mid = (low + high) / 2
= (1 + 1) / 2
= 1

low = mid + 1
= 0 + 1
= 1

Binary Search

Data : [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

Cari : 20

20 == 20

20 < 20

20 > 20

Implementation

```
int binarySearch(int arr[], int
data) {
    int low = 0, high = n;
    int mid = (low + high) / 2;
    while(low <= high) {
      [logic]
    }
    return -1;
}
```

```
[logic]
if(arr[mid] == data) {
  return mid;
} else if(data < arr[mid]) {
  high = mid - 1;
}else {
  low = mid + 1;
}
mid = (low + high) / 2;
[logic]
```