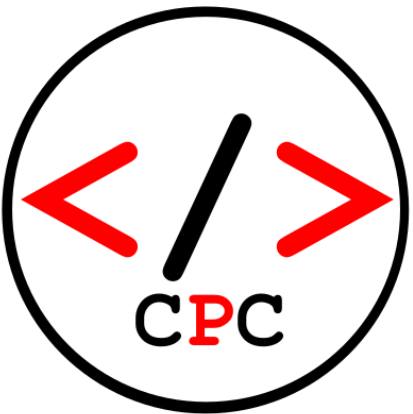


STL Part 1

Rosa A. S.



STL

- Fungsi Matematika
- Fungsi Algoritma pada Vector
- Set
- Pair



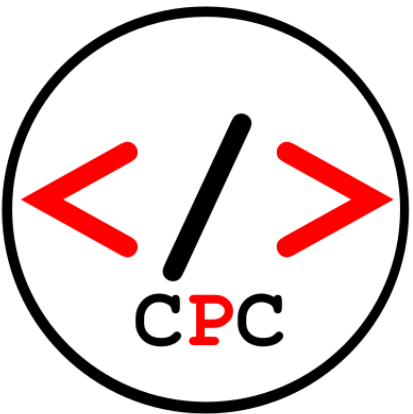
Fungsi Matematika

Fungsi/Prosedur	Keterangan
<code>int min(int, int)</code>	Menghasilkan nilai paling kecil diantara dua buah masukan parameter
<code>int max(int, int)</code>	Menghasilkan nilai paling besar diantara dua buah masukan parameter
<code>double sqrt(double x)</code>	Mengembalikan hasil akar kuadrat dari parameter masukan (parameter masukan harus bernilai 0 atau lebih besar)
<code>double ceil(double x)</code>	Mengembalikan nilai pembulatan ke atas/nilai lebih tinggi.



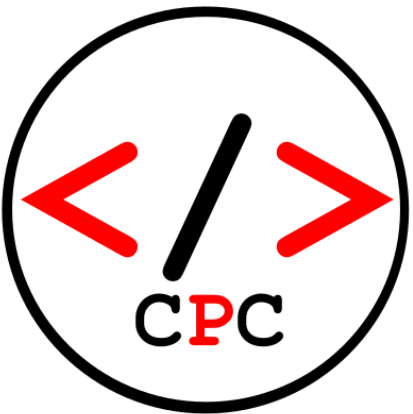
Fungsi Matematika

Fungsi/Prosedur	Keterangan
<code>int abs(int x)</code>	Menghasilkan bilangan mutlak dari
<code>double floor(double x)</code>	Mengembalikan nilai pembulatan ke bawah dari parameter
<code>double pow(double x, double y)</code>	Mengembalikan nilai pemangkatan dari parameter masukan, artinya x pangkat y



Fungsi Matematika

Fungsi/Prosedur	Keterangan
<code>double exp(double x)</code>	<p>Mengembalikan bilangan exponensial dari parameter, e adalah basis dari logaritma alami. Kadang-kadang disebut juga bilangan Euler sebagai penghargaan atas ahli matematika Swiss, Leonhard Euler.</p> <p>E sama dengan 2.7182818284590452354.</p>
<code>double log(double x)</code>	<p>Mengembalikan bilangan alami dari logaritma dari parameter masukan. Masukan harus lebih besar dari 0.</p>
<code>double log10(double x)</code>	<p>Mengembalikan bilangan alami dari logaritma berbasis 10 dari parameter masukan. Masukan harus lebih besar dari 0.</p>



Competitive **Programming** Club – Computer **Science** UPI

Algoritma di Vector: Pencarian

```
vs v;  
string s,sf;  
int i, n;  
  
cin >> n;  
  
for(i=0;i<n;i++){  
    cin >> s;  
    v.push_back(s);  
}  
  
for(i=0;i<n;i++) cout << v[i] << "\n";  
cin >> sf;  
  
if(find(v.begin(), v.end(), sf) != v.end()){  
    cout << "ada\n";  
}else{  
    cout << "tidak ada\n";  
}
```



Algoritma di Vector: Pengurutan

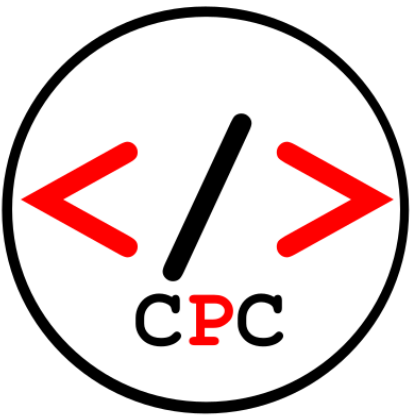
```
sort(v.begin(), v.end());
```

```
for(i=0;i<n;i++) cout << v[i] << "\n";
```

```
//besar ke kecil
```

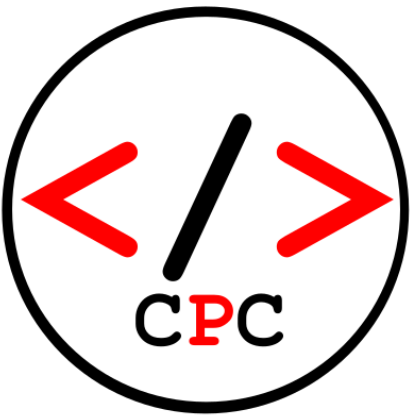
```
sort(v.begin(), v.end(), greater<string>());
```

```
for(i=0;i<n;i++) cout << v[i] << "\n";
```



Algoritma di Vector: Permutasi

```
do {  
    for(i=0;i<n;i++) cout << v[i] << " ";  
    cout << "\n";  
  
} while ( next_permutation (v.begin(),v.end()) );
```

STL: Set – menyimpan elemen unik

```
#include <bits/stdc++.h>

using namespace std;

int main() {

    ios::sync_with_stdio(0);
    cin.tie(0);

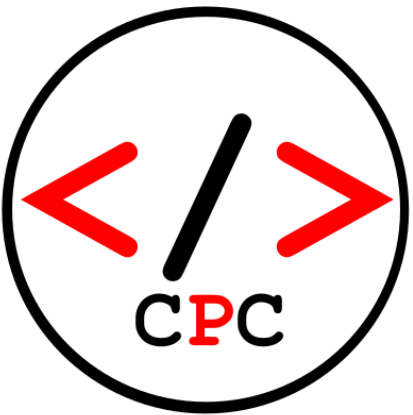
    string str;
    set<string> s;

    while(cin >> str) s.insert(str);

    set<string>::const_iterator it;

    for (it = s.begin(); it!=s.end(); it++) cout << *it << "\n";

    return 0;
}
```

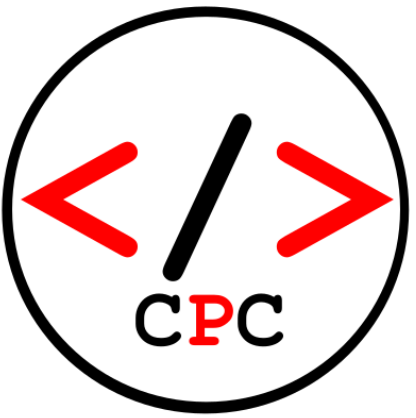


STL: Pair – menyimpan elemen yang berpasangan

```
pair<string, string> elmt;
```

```
cin >> elmt.first >> elmt.second;
```

```
cout << elmt.first << " " << elmt.second << "\n";
```



STL: Vector of Pair – menyimpan elemen yang berpasangan dalam Vector

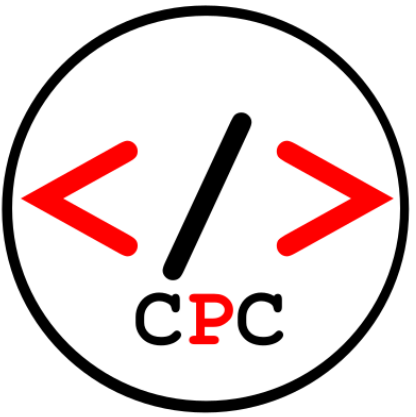
```
vector<pair<int, string>> vp;
```

```
int a, i;
```

```
string s;
```

```
while(cin >> a >> s) vp.push_back(make_pair(a, s));
```

```
for(i=0;i<vp.size();i++)cout << vp[i].first << " " << vp[i].second << "\n";
```



Next: STL

- Map
- Stack
- Queue
- Vector 2 Dimensi
- STL di dalam STL