

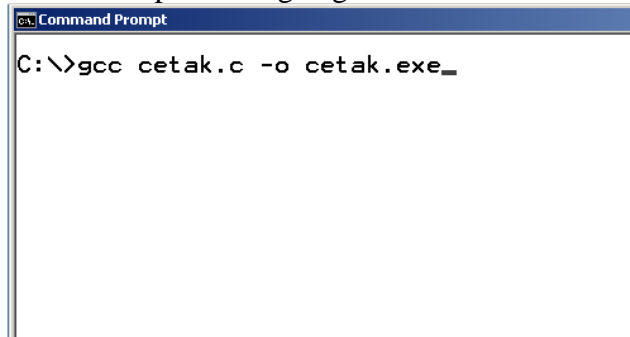
Command Line Argument

Dalam bahasa C, ada cara untuk mengirimkan argument/parameter dalam baris perintah ke program saat di eksekusi. Ketika fungsi `main` dipanggil dari prompt/dijalankan dari command line, fungsi tersebut dipanggil dengan 2 (dua) argumen. Pertama (biasanya disebut `argc`, untuk menghitung jumlah argument) yang merupakan jumlah dari argument yang dituliskan dalam command-line saat program dieksekusi, Kedua (`argv`, untuk argumen vector/array) yaitu suatu array pointer ke karakter yang mengandung argument string sejumlah parameter pertama `argc`. Kita dapat mengatur secara kustom dan menggunakan banyak pointer untuk memanipulasi karakter string ini. Perhatikan contoh di bawah ini.

Misalkan kita membuat program dengan nama *cetak.exe*, yang di hasilkan dari source code *cetak.c*, berikut ini :

```
#include <stdio.h>
/* cetak versi 1 */
int main(int argc, char *argv[])
{
    int i;
    for (i = 1; i < argc; i++)
        printf("%s%s", argv[i], (i < argc-1) ? " " : " ");
    printf("\n");
    return 0;
}
```

Saat di kompilasi dengan `gcc cetak.c -o cetak.exe`,



```
CA Command Prompt
C:\>gcc cetak.c -o cetak.exe_
```

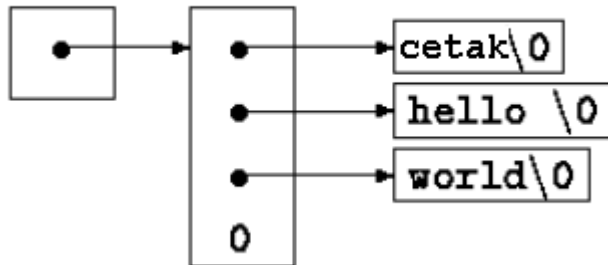
lalu di jalankan dari prompt/command line seperti di bawah ini :



```
CA Command Prompt
C:\>cetak Hello World
Hello World
C:\>_
```

Secara aturan maka, `argv[0]` merupakan nama dari program yang kita buat, dalam kasus ini `cetak.exe`, yang di panggil dari prompt, jadi `argc` **setidaknya bernilai 1**. Jika `argc` adalah 1, maka tidak ada argument lainnya setelah nama program saat kita panggil. Dalam contoh diatas, `argc` bernilai 3, masing-masing sebagai berikut `argv[0]`, `argv[1]`, dan `argv[2]` yang berisi "cetak", "Hello", and "World". Argumen pertama adalah `argv[1]` dan yang terakhir adalah `argv[argc-1]`. Gambar berikut akan lebih memudahkan penjelasan diatas.

argv:



Jadi dari potongan program diatas saat di panggil dengan `cetak hello world`, dapat di rangkum, sebagai berikut :

```
int main(int argc, char *argv[])
```

int argc : akan menghitung jumlah argument string setelah nama program, dalam contoh : `argc=3`, karena `cetak hello world` terdiri dari 3 string

char *argv[] : akan berisi argument string, termasuk nama programnya, dalam contoh maka isi dari `argv[]` menjadi `argv[2]`, karena index dalam bahasa C diawali dengan 0, maka `argv[0]=cetak`, `argv[1]=hello` dan `argv[2]=world`.

Yang perlu di perhatikan bahwa nama program atau argument pertama (`argv[0]`), tidak perlu ikut di proses, oleh karena itu, pernyataan `for (i=1; i<argc; i++)`, dimulai dari 1 dan bukan dari 0, dan akan di ulang sebanyak nilai `argc` (3).

Contoh-contoh berikut dapat di pakai sebagai referensi latihan :

1. Menghitung jumlah argumen

```
/*File : hitarg.c*/
#include <stdio.h>
#include <ctype.h>

int main(int argc, char *argv[])
{
    printf("Jumlah argumen program termasuk \n");
    printf("nama program adalah %d argumen\n",argc);
    getch();
    return 0;
}
```

```
}
```

2. Menghitung jumlah argument dan menampilkan argument string

```
/*File : hitarg1.c*/  
#include <stdio.h>  
#include <ctype.h>  
  
int main(int argc, char *argv[])  
{  
    int i;  
    printf("Jumlah argumen program termasuk \n");  
    printf("nama program adalah %d argumen\n",argc);  
    for(i=1;i<argc;i++)  
        printf("argumen program ke-%d adalah %s\n",i,argv[i]);  
    getch();  
    return 0;  
}
```

3. Membatasi jumlah argument yang boleh dikirimkan

```
/*File : batasarg.c*/  
#include <stdio.h>  
#include <ctype.h>  
  
int main(int argc, char *argv[])  
{  
    int i;  
    if ((argc<2)|| (argc>3)) {  
        printf("Program memerlukan argumen\n");  
        printf("Cara Pakai : batasarg <arg>\n");  
        printf("Dimana <arg> adalah argumen berjumlah maksimal 2\n");  
        printf("Contoh : batasarg satu dua\n");  
    }  
    else {  
        printf("Sempurna !\n");  
        printf("Program di panggil dengan %d argumen\n",argc-1);  
        for(i=1;i<argc;i++)  
            printf("argv[%d]=%s\n",i,argv[i]);  
    }  
    getch();  
    return 0;  
}
```

4. Mengkonversi argument string menjadi integer

```
/*File : tambah.c*/  
#include <stdio.h>  
#include <ctype.h>  
/*ctype.h adalah file header yang di pakai untuk memanggil  
fungsi konversi  
    atoi(string)==> mengkonversi alphanumeric menjadi integer
```

```

    atof(string)==> mengkonversi alphanumeric menjadi float
    dsb...
    silahkan cari petunjuk mengenai fungsi konversi dalam
    bahasa C, dalam library ctype.h, google lebih baik.
*/
int main(int argc, char *argv[])
{
    int a,b,c;
    if ((argc<=2)|| (argc>3)) {
        printf("Program memerlukan argumen\n");
        printf("Cara Pakai : batasarg <arg>\n");
        printf("Dimana <arg> adalah argumen berjumlah maximal 2\n");
        printf("Contoh : tambah 5 4\n");
    }
    else {
        /*Ingat argumen yang akan di proses adalah argumen
        ke-1 dan ke-2, yang ke-0 tidak diproses
        karena merupakan nama program itu sendiri*/

        /*Konversi tiap argumen string menjagi integer
        supaya dapat dilakukan operasi aritmatika
        dengan fungsi atoi()*/

        a=atoi(argv[1]);
        b=atoi(argv[2]);
        c=a+b;
        /*cetak output program*/
        printf("%d + %d = %d\n",a,b,c);
    }
    getch();
    return 0;
}

```

Latihan 1 :

Apakah output dari program di bawah ini, jika program di jalankan dengan command line sebagai berikut : **hitung 10 20 30 40**

```

/*File: hitung.c*/
#include<stdio.h>
#include<ctype.h>
int main(int argc, char *argv[]){
    int i;int sum=0;
    for (i = 1; i < argc; i++)
        sum=sum+atoi(argv[i]);
    printf("Total : %d\n",sum);
    return 0;
}

```

Latihan 2 :

Membuat program pemangkatan X^Y dimana X adalah basis dan Y adalah exponent, pemangkatan adalah mengalikan X sejumlah Y, atau mengalikan basis sejumlah exponennya. Misal:

5^2 adalah $5 \times 5 = 25$

2^8 adalah $2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 256$

Buatlah program yang memerlukan 2 argumen parameter tidak kurang dan tidak lebih, yaitu argument pertama argv[1] adalah basisnya dan argument kedua argv[2] adalah exponennya, sehingga dari command line akan dipanggil sebagai berikut :

```
C:\>pangkat 5 2
```

```
25
```

```
C:\>pangkat 2 8
```

```
256
```