[OS201][WEEK: 02][DEMO]

[CLASS: E][ID: 1806141132][Name: Ali Irsyaad Nursyaban]

Compile All File with make command :

```
ali.irsyaad@badak: ~/Demos/Week02/c-language
ali.irsyaad@badak:~/Demos/Week02/c-language$ make
gcc -std=gnu99 00-hallo.c -o 00-hallo
gcc -std=gnu99 01-printf-int.c -o 01-printf-int
gcc -std=gnu99 02-printf-etc.c -o 02-printf-etc
gcc -std=gnu99 03-loop.c -o 03-loop
gcc -std=gnu99 04-argc-argv.c -o 04-argc-argv
gcc -std=gnu99 05-envp.c -o 05-envp
gcc -std=gnu99 06-scan.c -o 06-scan
gcc -std=gnu99 07-status.c -o 07-status
gcc -std=gnu99 08-pointer.c -o 08-pointer
gcc -std=gnu99 09-function.c -o 09-function
gcc -std=gnu99 10-uts171.c -o 10-uts171
gcc -std=gnu99 11-uts172.c -o 11-uts172
gcc -std=gnu99 12-uts181.c -o 12-uts181
gcc -std=gnu99 13-uts182.c -o 13-uts182
gcc -std=gnu99 14-uts191.c -o 14-uts191
ali.irsyaad@badak:~/Demos/Week02/c-language$ ls -al
total 260
drwxr-xr-x 2 ali.irsyaad StafCS 4096 Feb 16 16:54 .
drwxr-xr-x 4 ali.irsyaad StafCS 4096 Jan 28 21:41 ..
-rwxr-xr-x 1 ali.irsyaad StafCS 8472 Feb 16 16:54 00-hallo
-rw-r--r-- 1 ali.irsyaad StafCS  514 Jan 28 21:41 00-hallo.c
-rwxr-xr-x 1 ali.irsyaad StafCS 8472 Feb 16 16:54 01-printf-int
-rw-r--r-- 1 ali.irsyaad StafCS 1114 Jan 28 21:41 01-printf-int.c
-rwxr-xr-x 1 ali.irsyaad StafCS 8528 Feb 16 16:54 02-printf-etc
-rw-r--r-- 1 ali.irsyaad StafCS 1764 Jan 28 21:41 02-printf-etc.c
-rwxr-xr-x 1 ali.irsyaad StafCS 8520 Feb 16 16:54 03-loop
-rw-r--r-- 1 ali.irsyaad StafCS 1030 Jan 28 21:41 03-loop.c
-rwxr-xr-x 1 ali.irsyaad StafCS 8528 Feb 16 16:54 04-argc-argv
-rw-r--r-- 1 ali.irsyaad StafCS 1669 Jan 28 21:41 04-argc-argv.c
```

* Compile c juga dapat dilakukan dengan command gcc tapi harus dilakukan dengan manual

Run 00-halo:

```
ali.irsyaad@badak: ~/Demos/Week02/c-language
ali.irsyaad@badak:~/Demos/Week02/c-language$ cat 00-hallo.c
/*
 * Copyright (C) 2016-2019 Rahmat M. Samik-Ibrahim
 * http://rahmatm.samik-ibrahim.vlsm.org/
 * This program is free script/software. This program is distributed in the
 * hope that it will be useful, but WITHOUT ANY WARRANTY; without even the
 * implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
 * REV04 Wed Jan 23 16:14:51 WIB 2019
 * REV03 Tue Aug 28 13:31:19 WIB 2018
 * START Tue Sep 13 11:51:21 WIB 2016
 */

#include <stdio.h>

void main(void) {
   printf("Hallo World!\n");
}

ali.irsyaad@badak:~/Demos/Week02/c-language$ ./00-hallo
Hallo World!
ali.irsyaad@badak:~/Demos/Week02/c-language$
```

Run 01-printf-int.c :

```
ali.irsyaad@badak: ~/Demos/Week02/c-language
   printf("(%%d)    ii = (%d)\n",    ii);
   printf("(%%5d)   ii = (%5d)\n",   ii);
   printf("(%%1d)   ii = (%1d)\n",   ii);
   printf("(%%-5d)  ii = (%-5d)\n",  ii);
   printf("(%%5.5d) ii = (%5.5d)\n", ii);
   printf("(%%5.3d) ii = (%5.3d)\n", ii);
   printf("(%%c)    ii = %c\n",      ii);
   printf("(%%x)    ii = %x\n",      ii);
   printf("(%%#x)   ii = %#x\n",     ii);
   printf("(%%X)    ii = %X\n",      ii);
   printf("(%%#X)   ii = %#X\n",     ii);
   printf("(%%o)    ii = %o\n",      ii);
   printf("(%%#o)   ii = %#o\n",     ii);
}

ali.irsyaad@badak:~/Demos/Week02/c-language$ ./01-printf-int
(%d)    ii = (74)
(%5d)   ii = (   74)
(%1d)   ii = (74)
(%-5d)  ii = (74   )
(%5.5d) ii = (00074)
(%5.3d) ii = (  074)
(%c)    ii = J
(%x)    ii = 4a
(%#x)   ii = 0x4a
(%X)    ii = 4A
(%#X)   ii = 0X4A
(%o)    ii = 112
(%#o)   ii = 0112
ali.irsyaad@badak:~/Demos/Week02/c-language$
```

Run 02-printf-etc.c :

```
ali.irsyaad@badak: ~/Demos/Week02/c-language

   char        ch='A';
   unsigned char uch = (unsigned char) -191;
   printf("char %c (%d) |||| unsigned char       %c (%u)\n", ch, ch, uch, uch);

#define STRING "123456789ABCDEF\n"

   char  dd[]=STRING;
   long  size=sizeof(STRING);
   printf(STRING);
   printf("One  more time:\n%sSIZE=%ld\n", dd, size);
   printf("dd[]= %c%c%c ... %c\n", dd[0],
          dd[1], dd[2], dd[size-3]);
}

ali.irsyaad@badak:~/Demos/Week02/c-language$ ./02-printf-etc
short     1 |||| unsigned short     65535
int       1 |||| unsigned           4294967295
long      1 |||| unsigned long      18446744073709551615 (0XFFFFFFFFFFFFFFFF)
long long 1 |||| unsigned long long 18446744073709551615 (0XFFFFFFFFFFFFFFFF)
float       123456792.000000 (1.234568e+08)
double      123456789.123457 (1.234568e+08)
long double 123456789.123457 (1.234568e+08)
char A (65) |||| unsigned char       A (65)
123456789ABCDEF
One  more time:
123456789ABCDEF
SIZE=17
dd[]= 123 ... F
ali.irsyaad@badak:~/Demos/Week02/c-language$
```

Run 03-loop.c :

```
ali.irsyaad@badak: ~/Demos/Week02/c-language
 *
 * The outer loop will execute the block
 * including the inner loop
 * 4 times as declared at the beginning in
 * variable OLOOP.
 *
 * The inner loop will print to console 3 times as declared
 * at the beginning in variable ILOOP
 */

ali.irsyaad@badak:~/Demos/Week02/c-language$ ./03-loop
==========
OL [0000]
    IL[0]
    IL[1]
    IL[2]
OL [0001]
    IL[0]
    IL[1]
    IL[2]
OL [0002]
    IL[0]
    IL[1]
    IL[2]
OL [0003]
    IL[0]
    IL[1]
    IL[2]
==========
ali.irsyaad@badak:~/Demos/Week02/c-language$
```

Run 04-argc-argv.c :

```
ali.irsyaad@badak: ~/Demos/Week02/c-language

#include <stdio.h>
void main(int argc, char *argv[]) {
   printf("The value of argc    is %d\n", argc);
   printf("===========================================\n");
   for (int ii=0; ii < argc; ii++) {
      printf("The value of argv[%d] is %s\n", ii, argv[ii]);
   }
   printf("===========================================\n");
}

ali.irsyaad@badak:~/Demos/Week02/c-language$ ./04-argc-argv A B C D
The value of argc    is 5
===========================================
The value of argv[0] is ./04-argc-argv
The value of argv[1] is A
The value of argv[2] is B
The value of argv[3] is C
The value of argv[4] is D
===========================================
ali.irsyaad@badak:~/Demos/Week02/c-language$
```

Run 05-envp.c :

```
ali.irsyaad@badak: ~/Demos/Week02/c-language
 * were entered on the command line when running this program
 * ENVP (program environment) is an array of strings that consists a list of the environment
 * variables of your shell
 * When you run this program, this program will print your program environtment on your console
 * This program will only print as many as 15 environtment (envp[0] - envp[14]) according to
 * the limit (MAXLOOP) and the environtment length is only 60 (MAXLENG)
 * Char 57 to 59 will be replaced with X (line 20-22) and the rest will not be printed
 */


ali.irsyaad@badak:~/Demos/Week02/c-language$ ./05-envp
============================================================
The value of envp[00] is dummy=dummy
The value of envp[01] is TERM=xterm-256color
The value of envp[02] is SHELL=/bin/bash
The value of envp[03] is HISTSIZE=1000
The value of envp[04] is SSH_CLIENT=152.118.25.2 59701 22
The value of envp[05] is SSH_TTY=/dev/pts/15
The value of envp[06] is NO_PROXY=152.118.0.0/16,127.0.0.0/8,::1,lXXX
The value of envp[07] is HISTFILESIZE=2000
The value of envp[08] is http_proxy=http://proxy.cs.ui.ac.id:8080/
The value of envp[09] is USER=ali.irsyaad
The value of envp[10] is LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:piXXX
The value of envp[11] is ftp_proxy=http://proxy.cs.ui.ac.id:8080/
The value of envp[12] is FTP_PROXY=http://proxy.cs.ui.ac.id:8080/
The value of envp[13] is ALL_PROXY=http://proxy.cs.ui.ac.id:8080/
The value of envp[14] is all_proxy=http://proxy.cs.ui.ac.id:8080/
============================================================
ali.irsyaad@badak:~/Demos/Week02/c-language$
```

Run 06-scan.c :

```
ali.irsyaad@badak: ~/Demos/Week02/c-language
void main(void) {
   int ii;
   char buf[10];
   printf("Silakan isi: ");
   scanf("%s",buf);
   printf("Isi buffer = %s\n",buf);
}

/*
 * TAKE NOTE(AII)
 * scanf() reads formatted data from standard input
 * and then writes the results into the arguments given
 *
 * the line scanf("%s",buf); will read string inputted by
 * user and write the value to variable buf
 */

ali.irsyaad@badak:~/Demos/Week02/c-language$ ./06-scan
Silakan isi: Hallo
Isi buffer = Hallo
ali.irsyaad@badak:~/Demos/Week02/c-language$
```

Run 07-status.c :

```
ali.irsyaad@badak: ~/Demos/Week02/c-language
void main(void) {
   printf("Process Identifier (PID) [%5.5d] -- Parent PID (PPID[%5.5d])\n", getpid(), getppid());
   sleep(1);
}

/*
 * TAKE NOTE(AII, MG)
 * In Linux, an executable stored on disk is called a
 * program,
 * and a program loaded into memory and running is
 * called a process.
 * When a process is started, it is given a unique
 * number called process ID (PID)
 * that identifies that process to the system
 *
 * Each process is assigned a parent process ID (PPID)
 * that tells which process started it.
 * The PPID is the PID of the process's parent.
 *
 * To see PID and PPID in current process you can use
 * getpid() and getppid()
 *
 * sleep() function delays program execution for a
 * given number of seconds
 */


ali.irsyaad@badak:~/Demos/Week02/c-language$ ./07-status
Process Identifier (PID) [18738] -- Parent PID (PPID[18369])
```

Run 08-pointer.c :

```
ali.irsyaad@badak: ~/Demos/Week02/c-language
   char*   ptr=&cca;
   printf ("Print  *ptr: %c\n", *ptr);
   ptr=&ccb;
   printf ("Print  *ptr: %c\n", *ptr);
   char    array[]="ABCDEF";
   printf ("Print array: %s\n", array);
   ptr=array;
   printf ("Print  *ptr: %c\n", *ptr);
   ptr=&array[0];
   printf ("Print  *ptr: %c\n", *ptr);
   ptr=&array[1];
   printf ("Print  *ptr: %c\n", *ptr);
   ptr=ptr+1;
   printf ("Print  *ptr: %c\n", *ptr);
   ptr++;
   printf ("Print  *ptr: %c\n", *ptr);
}

ali.irsyaad@badak:~/Demos/Week02/c-language$ ./08-pointer
Print   cca: a
Print   ccb: b
Print  *ptr: a
Print  *ptr: b
Print array: ABCDEF
Print  *ptr: A
Print  *ptr: A
Print  *ptr: B
Print  *ptr: C
Print  *ptr: D
ali.irsyaad@badak:~/Demos/Week02/c-language$
```

Run 09-function.c :

```
ali.irsyaad@badak: ~/Demos/Week02/c-language

void main(void) {
   int ii=1;
   printf("Addone1 %d -- %d \n", ii,  addone1(ii));
   printf("Addone1 %d -- %d \n", ii,  addone1(ii));
   printf("Addone1 %d -- %d \n", ii,  addone1(ii));
   printf("Addone2 %d -- %d \n", ii,  addone2(&ii));
   printf("Addone2 %d -- %d \n", ii,  addone2(&ii));
   printf("Addone2 %d -- %d \n", ii,  addone2(&ii));
   printf("Addone3 %d -- %d \n", ii, *addone3(&ii));
   printf("Addone3 %d -- %d \n", ii, *addone3(&ii));
   printf("Addone3 %d -- %d \n", ii, *addone3(&ii));
   system("echo This is echo from system. Cek demo.txt");
   system("echo This is echo from system. Cek demo.txt > demo.txt");
}

ali.irsyaad@badak:~/Demos/Week02/c-language$ ./09-function
Addone1 1 -- 2
Addone1 1 -- 2
Addone1 1 -- 2
Addone2 2 -- 2
Addone2 3 -- 3
Addone2 4 -- 4
Addone3 5 -- 5
Addone3 6 -- 6
Addone3 7 -- 7
This is echo from system. Cek demo.txt
ali.irsyaad@badak:~/Demos/Week02/c-language$ cat demo.txt
This is echo from system. Cek demo.txt
ali.irsyaad@badak:~/Demos/Week02/c-language$
```

Run 10-uts171.c :

```
ali.irsyaad@badak: ~/Demos/Week02/c-language
 * This program is free script/software. This program is distributed in the
 * hope that it will be useful, but WITHOUT ANY WARRANTY; without even the
 * implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
 * REV03 Wed Feb 27 19:10:33 WIB 2019
 * START Thu Mar 30 12:13:58 WIB 2017
 */

#include <stdio.h>

int tambah(int ii, int jj) {
        return ii + jj;
}

void main(void) {
   int ii = 4;
   printf("The return of tambah is %d\n", tambah(1,ii));
}

ali.irsyaad@badak:~/Demos/Week02/c-language$ ./10-uts171
The return of tambah is 5
ali.irsyaad@badak:~/Demos/Week02/c-language$
```

Run 11-uts172.c :

```
ali.irsyaad@badak: ~/Demos/Week02/c-language
char* getGlobal(void) {
   char* charPTR=&globalChar;
   printf("getGlobal1 %c\n", globalChar);
   *charPTR='b';
   printf("getGlobal2 %c\n", *charPTR);
   return charPTR;
}

void main (void) {
   char localChar='c';
   printf("==== main1 %c\n", localChar);
   localChar=*getGlobal();
   printf("==== main2 %c\n", localChar);
}

/*
 * TAKE NOTE (ZA)
 * Line 25: mengambil local char pada fungsi yaitu c
 * Line 26: memanggil fungsi getGlobal()
 * Line 17: mengambil global char yaitu a
 * Line 19: mengganti chartPTR dari a menjadi b
 * Line 27: menggambil local char terakhir yaitu b
 */

ali.irsyaad@badak:~/Demos/Week02/c-language$ ./11-uts172
==== main1 c
getGlobal1 a
getGlobal2 b
==== main2 b
ali.irsyaad@badak:~/Demos/Week02/c-language$
```

Run 12-uts181.c :

```
ali.irsyaad@badak: ~/Demos/Week02/c-language
ali.irsyaad@badak:~/Demos/Week02/c-language$ cat 12-uts181.c
/*
 * Copyright (C) 2018-2019 Rahmat M. Samik-Ibrahim
 * http://rahmatm.samik-ibrahim.vlsm.org/
 * This program is free script/software. This program is distributed in the
 * hope that it will be useful, but WITHOUT ANY WARRANTY; without even the
 * implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
 * REV02 Thu Jan 17 13:56:47 WIB 2019
 * REV01 Tue Sep 25 12:01:28 WIB 2018
 * START Mon Apr 30 10:32:01 WIB 2018
 */

#include <stdio.h>

void main(void) {
   char string[]="HALLO";
   printf("START\n");
   printf("%s\n",  string);
   printf("%c\n", *string);
   printf("%c\n",  string[1]);
   printf("STOP\n");
}

ali.irsyaad@badak:~/Demos/Week02/c-language$ ./12-uts181
START
HALLO
H
A
STOP
ali.irsyaad@badak:~/Demos/Week02/c-language$
```

Run 13-uts182.c :

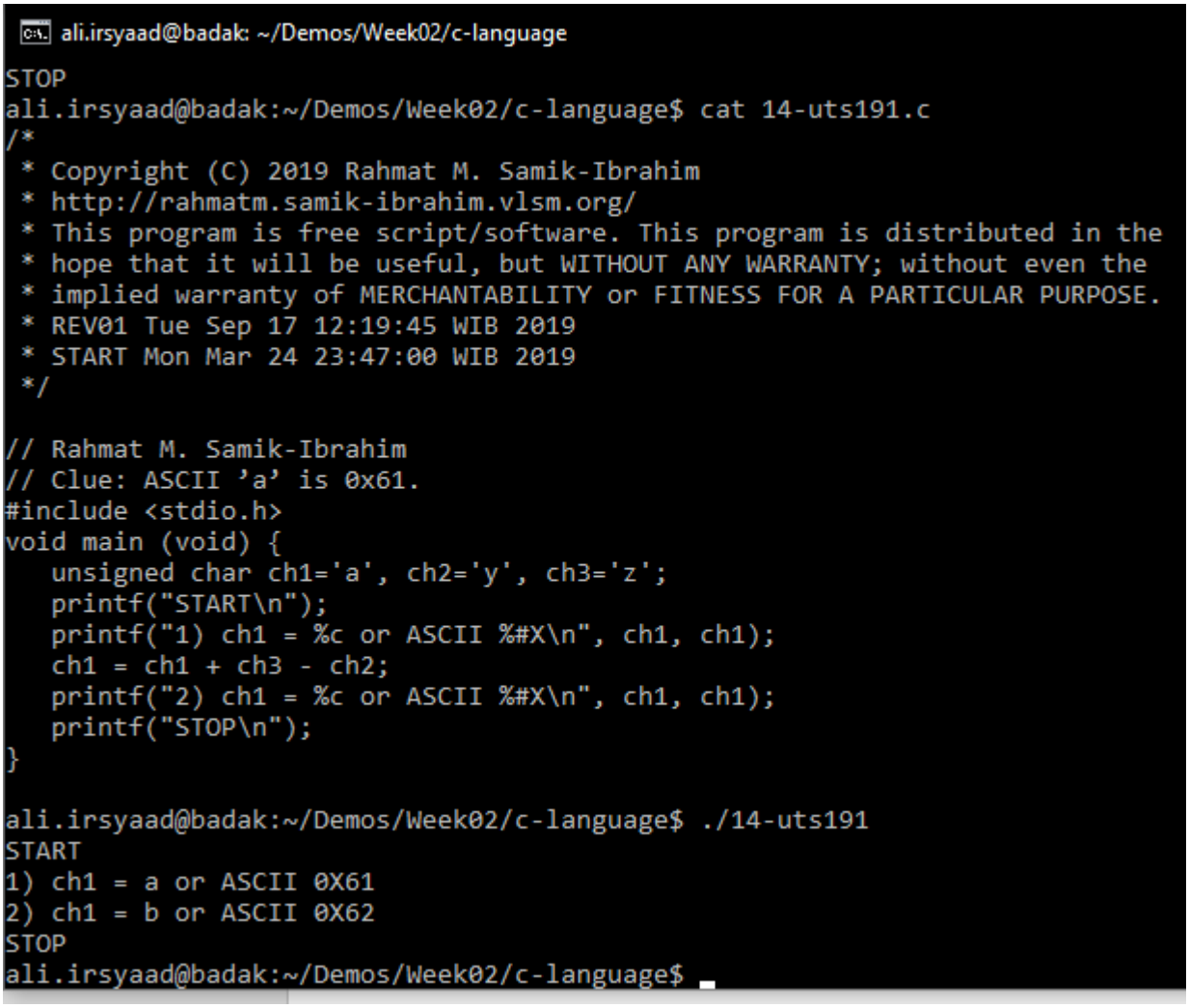
```
ali.irsyaad@badak: ~/Demos/Week02/c-language
 * This program is free script/software. This program is distributed in the
 * hope that it will be useful, but WITHOUT ANY WARRANTY; without even the
 * implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
 * REV01 Thu Jan 17 13:51:44 WIB 2019
 * START Xxx Xxx XX XX:XX:XX XXX XXXX
 * NOTE: ASCII 61H = a; 62H = b
 */

#include <stdio.h>

void main(void) {
   unsigned int   ii='a';
   unsigned char  ch='b';
   unsigned char* st="dcba";
   printf("START\n");
   printf(" ii    = %X or %c\n",   ii,  ii);
   printf(" ch    = %X or %c\n",   ch,  ch);
   printf("*st    = %X or %c\n",  *st, *st);
   printf(" st[2] = %X or %c\n", st[2], st[2]);
   printf("STOP\n");
}

ali.irsyaad@badak:~/Demos/Week02/c-language$ ./13-uts182
START
 ii    = 61 or a
 ch    = 62 or b
*st    = 64 or d
 st[2] = 62 or b
STOP
ali.irsyaad@badak:~/Demos/Week02/c-language$
```

Run 14-uts191.c :



Kesimpulan :

C adalah bahasa yang masih sangat sederhana dan digunakan untuk membuat program2 dasar yang kita gunakan pada perkuliahan OS ini. Maka dari itu belajar C sangatlah penting agar kita dapat membuat program kita sendiri. Cara compile C pada dasarnya adalah menggunakan command gcc, MakeFile hanyalah sebuah configurasi otomatisasi yang didalamnya kita buat script untuk run nanti yang dimana dalam hal ini adalah command gcc untuk compile program c kita.