**Pernyataan *goto***merupakan instruksi untuk mengarahkan eksekusi ke pernyataan yang diawali dengan suatu label. Label sendiri berupa suatu pengenal *identifier* yang diikuti dengan tanda titik dua (:)

Contoh pemakaian *goto* ditujukan pada program di bawah ini:

Pernyataan

goto cetak;

Mengisyaratkan agar eksekusi dilanjutkan ke pernyataan yang diawali dengan label

cetak:

Pernyataan

if (++pencacah <=10)

goto cetak;

#include <stdio.h>

int main() {

int i = 1;

cetak: // Label 'cetak'

printf("Iterasi ke-%d\n", i);

i++;

if (i <= 10) {

goto cetak; // Lompat kembali ke label 'cetak'

}

printf("Selesai\n");

return 0;

}

Mempunyai arti :

* Naikkan nilai **pencacah** sebesar 1
* Kemudian, jika **pencacah** kurang dari atau sama dengan 10 maka eksekusi menuju ke label **cetak.**

Penerapan *goto* biasanya dilakukan pada *loop* di dalam *loop (nested loop)*, dengan tujuan memudahkan untuk keluar dari *loop* terdalam menuju ke pernyataan yang terletak di luar *loop* terluar.

**Menggunakan *exit* Untuk menghentikan eksekusi program**

Suatu eksekusi program dapat dihentikan (secara normal) melalui pemanggilan fungsi *exit()*. Hal ini biasa dilakukan, jika di dalam suatu eksekusi terdapat suatu kondisi yang tak dikehendaki. Prototipe dari fungsi *exit()* didefinisikan pada file **stdlib.h**, yang memiliki deklarasi sebagai berikut:

void exit(int status);

Menurut kebiasaan, nilai nol diberikan pada argumen *exit()* untuk menunjukkan program yang normal. Sedangkan untuk menunjukkan kesalahan, nilai yang diberikan pada argumen fungsi diisi dengan nilai bukan-nol. Pada contoh program berikut, eksekusi program akan dihentikan hanya jika tombol ‘x’ ditekan

/\* File program : keluar.c

Pemakaian exit() untuk menghentikan eksekusi program \*/

#include <stdio.h>

#include <stdlib.h>

main()

{  
 char kar;

printf(“Tekanlah X untuk menghentikan program.\n”);

for ( ; ;)

{

while ((kar = getchar()) == ‘X’)

exit(0);

}

}

#include <stdio.h>

#include <stdlib.h>

int main() {

char karakter;

printf("Tekan 'X' untuk menghentikan program.\n");

for ( ; ;)

while ((karakter = getchar()) == 'X') {

exit(0);

}

}

**Exercise**

**Soal:** Buat program yang mencetak angka dari 1 hingga 5. Namun, jika angka adalah 3, gunakan goto untuk melewatkan pencetakan angka tersebut dan langsung melanjutkan ke angka berikutnya.

#include <stdio.h>

/\*

\*Bayangkan goto seperti seorang pemandu yang berkata, "Lompat ke sini,

\*lewati bagian ini!" Jadi, kita bisa membuat program agar melewatkan

\*pencetakan angka 3 dan melanjutkan ke angka 4.

\*/

int main(void) {

int i = 1;

mulai: // Label untuk mulai cetak angka

if (i > 5) {

goto udahan; // Jika i lebih besar dari 5, lompat ke akhir program

}

if (i == 3) {

i++; // Naikkan angka ke 4 sebelum lompat

goto mulai; // Lompat ke label 'start', lewati pencetakan 3

}

printf("%d\n", i); // Cetak angka jika tidak sama dengan 3

i++;

goto mulai; // Kembali ke 'start' untuk angka berikutnya

udahan: // Label untuk mengakhiri program

printf("Program selesai!\n");

return 0;

}

**Interpretasi** program ini mencetak angka 1, 2, 4, dan 5. Ketika i bernilai 3, goto start; digunakan untuk melompat kembali ke awal, melewatkan printf untuk angka 3.

**Soal:** Program Login menggunakan goto untuk mengulang input password jika salah

Interpretasi:

* Program mensimulasikan sistem login sederhana seperti di media sosial atau email
* Cara kerja:
  + User diminta memasukkan password
  + Jika password salah (tidak sama dengan " #admin123"):
    - Tampil pesan "Password salah!"
    - Program kembali ke label 'login' untuk meminta input ulang
  + Jika password benar:
    - Tampil pesan "Login berhasil!"
    - Program selesai

**Clue:**

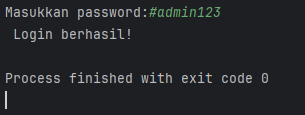
Menggunakan logika if(strcmp(password, "#admin123") != 0):

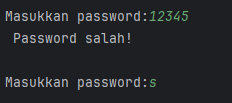
Pernyataan ini menggunakan fungsi strcmp() untuk membandingkan isi variabel password dengan string "#admin123".

!= 0 berarti kondisi akan terpenuhi (true) jika hasil dari strcmp() bukan 0, yang berarti password tidak sama dengan "#admin123".

Dengan kata lain, blok kode di dalam if hanya akan dijalankan jika password bukan "#admin123".

Output:





**2:**

#include <stdio.h>

#include <stdlib.h>

int main() {

int pilihan;

mulai\_sarapan:

printf("\n=== MENU SARAPAN ===\n");

printf("1. Nasi Goreng\n");

printf("2. Roti Bakar\n");

printf("3. Sereal\n");

printf("4. Tidak Mau Sarapan\n");

printf("Mau sarapan apa? (1-4): ");

scanf("%d", &pilihan);

if(pilihan == 4) {

printf("Kamu harus sarapan! Ayo pilih makanan!\n");

goto mulai\_sarapan; // Kembali ke menu sarapan

}

if(pilihan < 1 || pilihan > 4) {

printf("Menu tidak ada! Program berhenti!\n");

exit(1); // Program berhenti jika pilihan tidak valid

}

printf("Selamat menikmati sarapanmu!\n");

return 0;

}

**Interpretasi:**

Soal