Analisis Program Aplikasi Kasir 911 MART

1. Source Code

```c
#include <stdio.h>
#include <string.h>

char p[5];
typedef char *s;
    s nama[13]={"0", "Ayam", "Susu", "Buah", "Ikan", "Kaos", "Celana", "Kemeja", "Switer", "Mug", "Wajan", "Mangkuk", "Piring"};
typedef char *s1;
    s1 setNama[100];
int i, j, barang, pembayaran, kembali, kurang, setKode, tsh=0, value=0, jumlah[100], totalharga[100], setjumlah[100], setHarga[100], getKode[100];
int size = sizeof(nama)/sizeof(nama[0]);
int kode[]={0, 9110101, 9110102, 9110103, 9110104, 9110201, 9110202, 9110203, 9110204, 9110301, 9110302, 9110303, 9110304};
int harga[]={0, 35000, 11000, 15000, 10000, 55000, 65000, 80000, 110000, 20000, 25000, 10000, 8000};
int main () {

    printf("\t\t\t=========================================\n");
    printf("\t\t\t|       999999       111       111      |\n");
    printf("\t\t\t|      99    99     1111      1111      |\n");
    printf("\t\t\t|      99    99    11111     11111      |\n");
    printf("\t\t\t|       9999999       11        11      |\n");
    printf("\t\t\t|            99       11        11      |\n");
    printf("\t\t\t|           999       11        11      |\n");
    printf("\t\t\t|      9999999        11        11      |\n");
    printf("\t\t\t-----------------------------------------\n");
    printf("\t\t\t================= MART =================\n");
    printf("\t\t\t-----------------------------------------\n");
    printf("\t\t\t|                                       |\n");
    printf("\t\t\t-----------------------------------------\n");
    printf("\t\t\t  Daftar Barang Yang Tersedia Di 911 MART  \n");
    printf("\t\t\t=========================================\n");
    printf("\t\t\t|  Kode  | Nama Barang | Harga Barang(Rp) |\n");
    printf("\t\t\t|--------|-------------|------------------|\n");
    for(i=1; i<13 ; i++) {
        printf("\t\t\t|%d |%12s |Rp.%11d.00 |\n", kode[i], nama[i], harga[i]);
    }
    printf("\t\t\t=========================================\n");

    awal:
        printf("\t\t\t Masukkan banyak barang\t\t: ");
        scanf("%i",&barang);
```

2. Contoh Output

```
                    -----------------------------------------
                      Daftar Barang Yang Tersedia Di 911 MART
                    =========================================
                    |  Kode  | Nama Barang | Harga Barang(Rp) |
                    |--------|-------------|------------------|
                    |9110101 |        Ayam |Rp.      35000.00 |
                    |9110102 |        Susu |Rp.      11000.00 |
                    |9110103 |        Buah |Rp.      15000.00 |
                    |9110104 |        Ikan |Rp.      10000.00 |
                    |9110201 |        Kaos |Rp.      55000.00 |
                    |9110202 |      Celana |Rp.      65000.00 |
                    |9110203 |      Kemeja |Rp.      80000.00 |
                    |9110204 |      Switer |Rp.     110000.00 |
                    |9110301 |         Mug |Rp.      20000.00 |
                    |9110302 |       Wajan |Rp.      25000.00 |
                    |9110303 |     Mangkuk |Rp.      10000.00 |
                    |9110304 |      Piring |Rp.       8000.00 |
                    =========================================
                     Masukkan banyak barang         : 2

                     Masukkan kode barang ke-1      : 9110103
                     Masukkan jumlah barang         : 4


                     Masukkan kode barang ke-2      : 9110304
                     Masukkan jumlah barang         : 5

                     Total harga keseluruhan        : Rp. 100000
                     Masukkan jumlah pembayaran     : Rp. 100000
                    =========================================
                     Daftar belanja anda:
                     1. 4 Buah (@ Rp. 15000)        : Rp. 60000
                     2. 5 Piring (@ Rp. 8000)       : Rp. 40000
                     Total harga                    : Rp. 100000
                     Pembayaran                     : Rp. 100000
                     Kembalian                      : Rp. 0

                     Ulangi (y/t): t
                    =========================================
                    |               Terimakasih             |
                    |             Telah berbelanja          |
                    |                   Di                  |
                    |                 911 MART              |
                    =========================================
--------------------------------
Process exited after 36.82 seconds with return value 0
```

3. Analisis

```
#include <stdio.h>
#include <string.h>
```

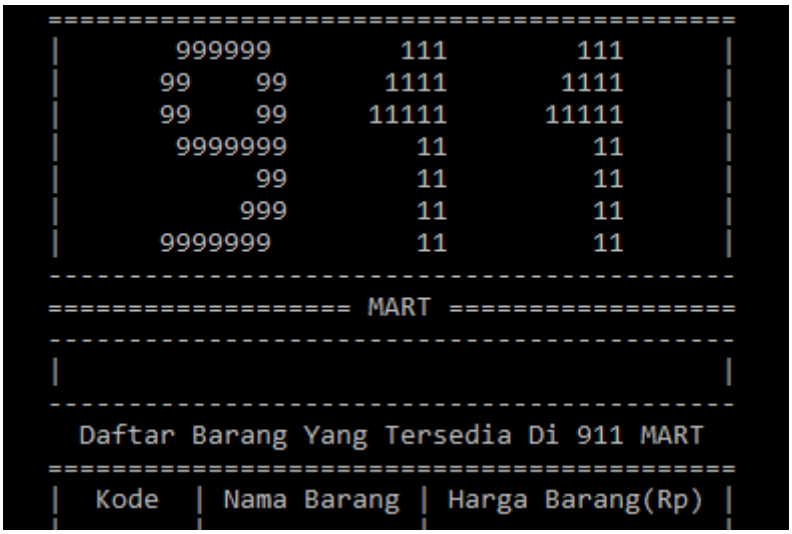
Pada baris ini maksudnya adalah memberitahukan *compiler* agar menggunakan *file header* `stdio.h` dan **`string.h`** yang ada pada direktori file-file *header*. `Stdio.h` berfungsi untuk mengeluarkan nilai/data mendefinisikan sebuah *input*/*output* bertipe data integer, dsb. Sedangkan `string.h` memiliki fungsi untuk menangani pemrosesan *string* atau *substring*.

```
char p[5];
typedef char *s;
    s nama[13]={"0", "Ayam", "Susu", "Buah", "Ikan", "Kaos", "Celana", "Kemeja", "Switer", "Mug", "Wajan", "Mangkuk", "Piring"};
typedef char *s1;
    s1 setNama[100];
int i, j, barang, pembayaran, kembali, kurang, setKode, tsh=0, value=0, jumlah[100], totalharga[100], setjumlah[100], setHarga[100], getKode[100];
int size = sizeof(nama)/sizeof(nama[0]);
int kode[]={0, 9110101, 9110102, 9110103, 9110104, 9110201, 9110202, 9110203, 9110204, 9110301, 9110302, 9110303, 9110304};
int harga[]={0, 35000, 11000, 15000, 10000, 55000, 65000, 80000, 110000, 20000, 25000, 10000, 8000};
int main () {
```

Baris ini merupakan tipe data. Tipe data adalah jenis nilai yang akan tersimpan dalam *variable*. `Char`: adalah tipe data yang berisi 1 huruf atau 1 karakter. Di atas menunjukkan `p[5]` dengan `[5]` didefinisikan sebagai ukuran maksimum. `typedef` adalah perintah atau *keyword* bahasa C yang dipakai untuk memberikan nama lain atau alias dari tipe data. Sebagai contoh, di atas dibuat tipe data `s` yang merupakan alias dari tipe data `char*`. Pernyataan `char *s` membuat literal string. Maka pada saat program dijalankan, akan muncul nilai dari variabel yang ditunjuk poleh pointer s. `nama [13]` merupakan data array dengan tipe data **char** dan memiliki ukuran maks 13. Elemen data arraynya adalah yang ada di sebelah kanan *variable* dengan diawali '{' dan diakhiri '}'. Tipe data `int` adalah tipe data yang berupa angka. `int size = sizeof(nama)/sizeof(nama[0]);` merupakan tipe data `int` untuk mendefinisikan `size` sebagai: ukuran *variable* `nama` dibagi ukuran *variable* `nama[0]`. `Sizeof()` adalah fungsi yang mengambil ukuran memori dari array. `main ()` adalah fungsi utama yang sebagai *starting point* di program c. `main ()` nanti akan dikembalikan sehingga akan disertakan `return 0` di akhir fungsi. Di dalam fungsi `main ()` berisi program yang akan dijalankan yang akan dibahas di bawah ini.

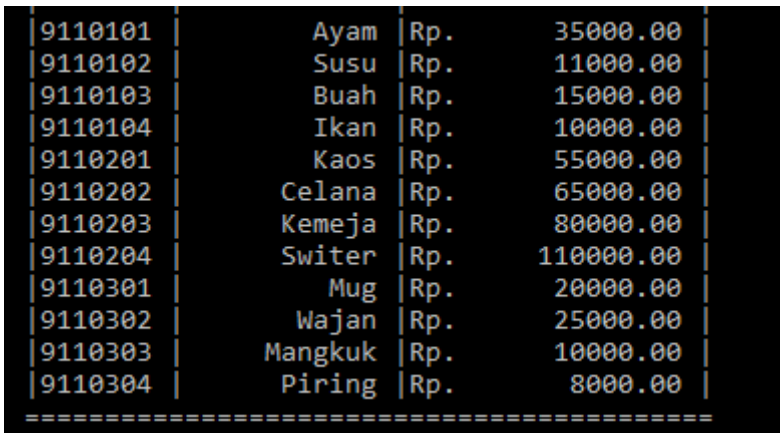
```
    printf("\t\t\t=========================================\n");
    printf("\t\t\t|       999999        111        111      |\n");
    printf("\t\t\t|      99    99      1111       1111      |\n");
    printf("\t\t\t|      99    99     11111      11111      |\n");
    printf("\t\t\t|       9999999        11         11      |\n");
    printf("\t\t\t|            99        11         11      |\n");
    printf("\t\t\t|           999        11         11      |\n");
    printf("\t\t\t|     9999999         11         11      |\n");
    printf("\t\t\t-----------------------------------------\n");
    printf("\t\t\t================== MART =================\n");
    printf("\t\t\t-----------------------------------------\n");
    printf("\t\t\t|                                       |\n");
    printf("\t\t\t-----------------------------------------\n");
    printf("\t\t\t  Daftar Barang Yang Tersedia Di 911 MART  \n");
    printf("\t\t\t=========================================\n");
    printf("\t\t\t|  Kode  | Nama Barang | Harga Barang(Rp) |\n");
    printf("\t\t\t|--------|-------------|------------------|\n");
    for(i=1; i<13 ; i++) {
        printf("\t\t\t|%d |%12s |Rp.%11d.00 |\n", kode[i], nama[i], harga[i]);
    }
    printf("\t\t\t=========================================\n");
```

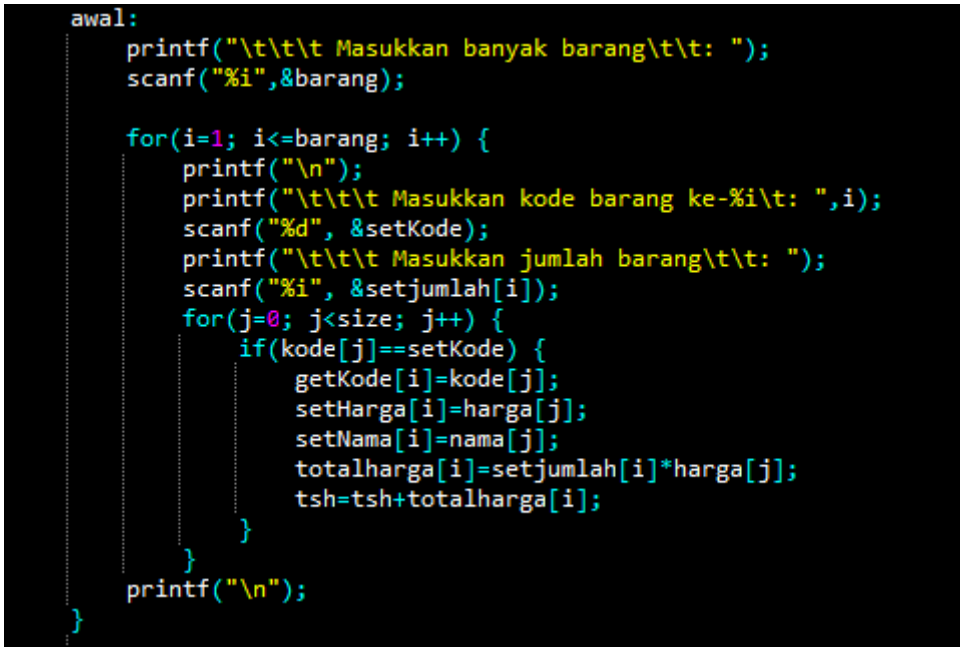
Fungsi `printf()` merupakan fungsi untuk menampilkan output ke layar komputer. Fungsi ini terdapat pada library `stdio.h`. Oleh sebab itu, ketika kita diharuskan untuk menuliskan `#include <stdio.h>` di bagian atas program agar bisa menggunakan fungsi ini. Cara penulisannya adalah `printf("format",...)`. `"format"` adalah sebuah teks (*string*) untuk ditampilkan. Lalu tanda `...` akan berisi sebuah variabel atau nilai untuk ditampilkan berdasarkan format yang diberikan pada teks `"format"`. Dengan `"\t"` melakukan tab dan `"\n"` melakukan enter pada baris. Dimasukkan sedemikian hingga memunculkan *output* seperti di bawah ini.




```
for(i=1; i<13 ; i++) {
      printf("\t\t\t|%d |%12s |Rp.%11d.00 |\n", kode[i], nama[i], harga[i]);
      }
```

Untuk `for` adalah pengulangan dengan `i=1` adalah awal, `i<13` berarti pengulangan sampai 13, dan `i++` menandakan i akan terus bertambah 1. Pengulangan akan mencetak *output* elemen 1-13 dari array `kode[i]` yang dimunculkan dengan `%d`, `nama[i]` dengan `%12s`, dan `harga[i]` dengan `%11d`. `%s` menampilkan *string*. `%d` menampilkan angka. Angka di sebelah kanan `%` menentukan jumlah *string* atau angka yang akan ditampilkan. Akan memunculkan *output* seperti ini.





Bagian dari `awal:. scanf("%i",&barang);.` Fungsi `scanf()` adalah fungsi untuk mengambil *input* dari *keyboard*. Fungsi ini memiliki format seperti fungsi `printf()`. Mengambil inputan untuk mengidenftifikasi banyak barang. Lalu melakukan pengulangan bagian `printf("\n");` sampai di bagian `printf("\n");` yang berada di bawah pada gambar di atas sejumlah `%i` yang diinputkan. `scanf("%d", &setKode);` akan memasukkan inputan yang merupakan anggota dari *variable* `setKode`.

```
for(j=0; j<size; j++) {
      if(kode[j]==setKode) {
            getKode[i]=kode[j];
            setHarga[i]=harga[j];
            setNama[i]=nama[j];
            totalharga[i]=setjumlah[i]*harga[j];
            tsh=tsh+totalharga[i];
```

Bagian di atas dapat dibaca begini. Akan melakukan pengulangan elemen dari 0 sampai ukuran dari suatu array. Saat `setKode` sama dengan anggota ke-j dari `kode[j]`, dengan sedemikian rupa dimana nanti akan memunculkan anggota ke-j dari `nama[j], kode[j],` dan `harga[j]`. Total harga akan dimunculkan dari `totalharga[i]=setjumlah[i]*harga[j]; tsh=tsh+totalharga[i];`

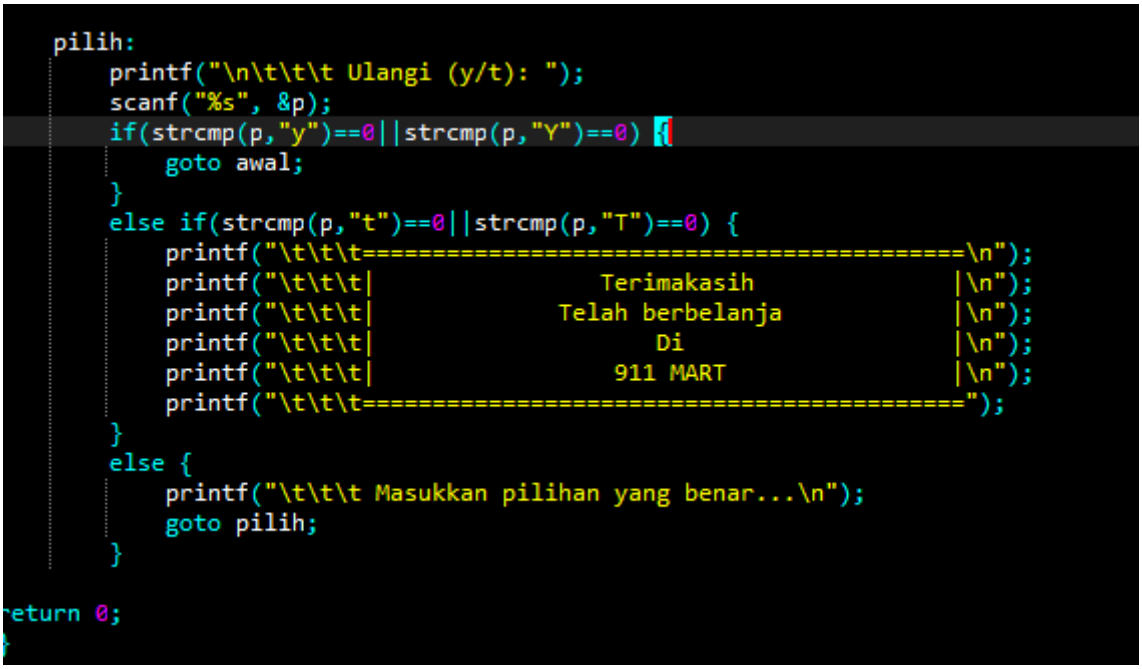
```
    printf("\t\t\t Total harga keseluruhan\t: Rp. %i\n", tsh);
    printf("\t\t\t Masukkan jumlah pembayaran\t: Rp. ");
    scanf("%i", &pembayaran);
    kembali=pembayaran-tsh;
    printf("\t\t\t=======================================\n");

    printf("\t\t\t Daftar belanja anda:\n");
    i=1;
    do {
        printf("\t\t\t %i. %i %s (@ Rp. %i)\t: Rp. %i\n", i, setjumlah[i], setNama[i], setHarga[i], totalharga[i]);
        i++;
    }
    while (i<=barang);

    if(pembayaran>=tsh) {
        value=1;
    }
    else {
        value=2;
    }

    switch (value) {
        case 1:
            printf("\t\t\t Total harga\t\t\t: Rp. %i\n", tsh);
            printf("\t\t\t Pembayaran\t\t\t: Rp. %i\n", pembayaran);
            printf("\t\t\t Kembalian\t\t\t: Rp. %i\n", kembali);
            break;
        case 2:
            kurang=-1*kembali;
            printf("\t\t\t Total harga\t\t\t: Rp. %i\n", tsh);
            printf("\t\t\t Pembayaran\t\t\t: Rp. %i\n", pembayaran);
            printf("\t\t\t Kekurangan\t\t\t: Rp. %i\n", kurang);
            break;
            default: printf("\n");
    }
```

```
scanf("%i", &pembayaran);
        kembali=pembayaran-tsh;
```

"%i" akan memunculkan jumlah pembayaran. Dan kembalian akan dihitung dengan `kembali=pembayaran-tsh;` . Pengulangan untuk `printf("\t\t\t %i. %i %s (@ Rp. %i)\t: Rp. %i\n", i, setjumlah[i], setNama[i], setHarga[i], totalharga[i]);` sejumlah *variable* `barang`, dimana `barang` menunjukkan banyak jenis barang. `%i. %i %s (@ Rp. %i)\t: Rp. %i\n"` masing masing akan memunculkan nomer, jumlah barang, harga per barang, dan total harga.

```
if(pembayaran>=tsh) {
        value=1;
}
else {
        value=2;
}
```

Jika pembayaran lebih dari total seluruh harga maka masuk ke `value=1` yanga akan mencetak kembalian. Jika pembayaran kurang dari total seluruh harga maka masuk ke `value=2` yang mencetak kekurangan. Kekurangan diambil dari -1 dikali kembalian. `Switch` berfungsi untuk menjalankan suatu perintah jika suatu `case` terpenuh i. Contohnya di atas saat `value=1`, `case '1'` akan aktif karena fungsi `switch` menjalankan *variable* `value` yang akan memuncukan apa yang ada di `printf`. Dicantumkan `break;` untuk mengakhiri `case`.



Masuk ke bagian `pilih:` yang menjadi bagian akhir. `Printf` akan memuncukan `Ulang (y/t)`. y atau t akan di `scanf` dan masuk ke *variable* p. Kasus `if`, `else if`, dan `else` nya adalah jika terbaca *string* ‘y’ atau ‘Y’ maka program akan kembali ke bagian awal:, jika terbaca *string* ‘t’ atau ‘T’ maka akan memunculkan apa yang ada di dalam `printf` di atas., dan jika terbaca *string* lain maka kembali ke `pilih:` karena bukan *string* yang diminta. Fungsi `strcmp()` digunakan untuk membandingkan *string* dengan *string* lainnya dan akan menghasilkan nilai 0 jika yang dibandingkan sama. `return 0` untuk mengembalikan ke fungsi `main()`.