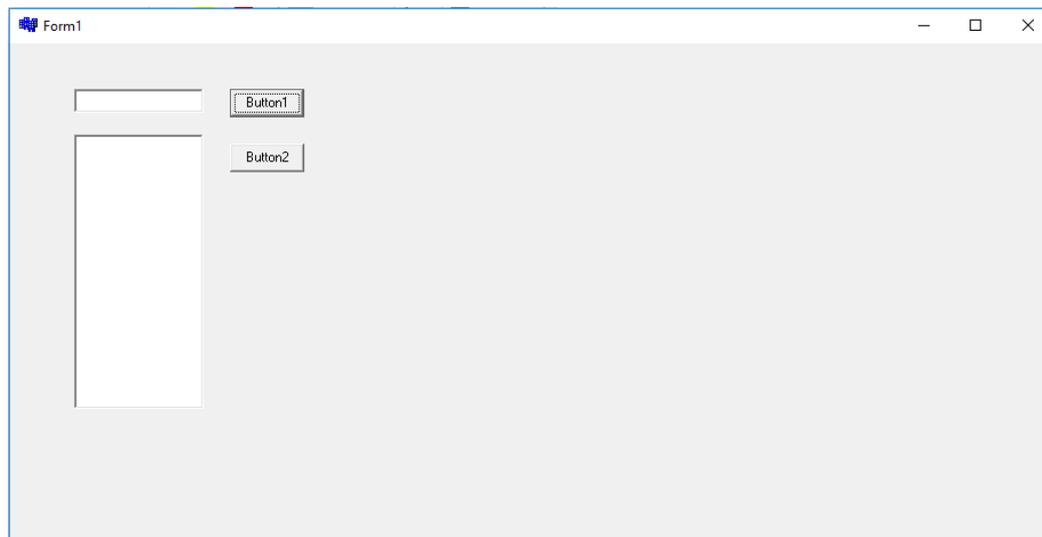


**PROGRAM SIMULASI STACK MENGGUNAKAN ARRAY
PADA BORLAND CBUILDER**

**SURYA AFNARIUS
SISTEM INFORMASI
UNIVERSITAS ANDALAS**

PROGRAM SIMULASI STACK MENGGUNAKAN ARRAY PADA BORLAND CBUILDER

Pada program simulasi stack ini, dilakukan penyusunan tumpukan piring/kertas yang diwakili dengan angka. Piring pertama yang dicuci diletakkan pada bagian dasar/bawah. Piring selanjutnya yang dicuci diletakkan diatas piring pertama tadi. Piring berikutnya yang dicuci akan berada diatas piring sebelumnya. Jika dilakukan pengambilan piring untuk dipakai, maka piring yang diambil adalah piring yang paling atas.



- a. Terdapat 1 EditText untuk menginputkan angka
- b. Terdapat 2 Button :
Button1 berfungsi untuk menambahkan inputan angka pada tumpukan
Button2 berfungsi untuk mengeluarkan angka teratas pada tumpukan
- c. Terdapat 1 ListBox untuk menampilkan angka pada tumpukan

Program ciptaan (Simulasi Stack Menggunakan Array) yang akan didaftarkan adalah:

```
//Variable Global

int kertas[10]={0,0,0,0,0,0,0,0,0,0};

» tempat untuk menyimpan tumpukan piring/kertas, inialisasi 0
diartikan sebagai belum ada piring/kertas yang masuk atau tempat
masih kosong.

int indeks=-1;

» sebagai indeks dari piring/kertas yang masuk.

//Button1 "Piring/Kertas Masuk"

if (indeks<9) { //Sebagai Batasan piring/kertas yang masuk

    ListBox1->Clear();

    indeks=indeks+1; //perubahan indeks ketika

    kertas[indeks]=StrToInt(Edit1->Text);
//piring/kertas masuk

    int w=9;

    while (w>=0) {

        ListBox1->Items->Add(kertas[w]); //untuk
menampilkan tumpukan piring/kertas

        w--; // kondisi peubah dalam perulangan

    }

}

else { // menampilkan pesan jika piring/kertas melebihi
batas

    ShowMessage ("Penuh");

}

//Button2 "Ambil Piring/Kertas"

if (indeks>=0) { // mengecek apakah jumlah piring/kertas
masih ada

    kertas[0]=0; // mengambil tumpukan piring/kertas
yang paling bawah

    indeks--; //mengurangi jumlah piring/kertas yang
tersedia

    ListBox1->Clear();
```

```

        int w=9;

        int i=0;

        while(i<=indeks+1){//perulangan dimulai dari yang paling atas

            kertas[i]=kertas[i+1];//agar tumpukan yang diatas jatuh ke bawah

            i++;// kondisi peubah perulangan

        }

        while(w>=0){//menampilkan piring/kertas dari yang paling atas

ListBox1->Items->Add(kertas[w]);

            w=w-1;//kondisi peubah perulangan

        }

    }

    else{//menampilkan pesan jika piring/kertas tidak tersedia/kosong

        ShowMessage("Piring/Kertas Kosong");

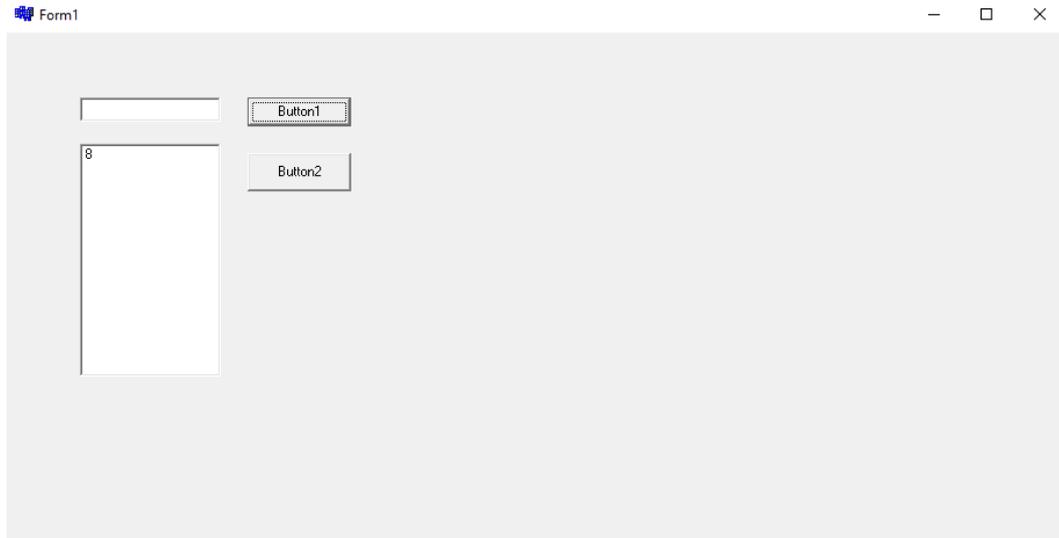
    }

```

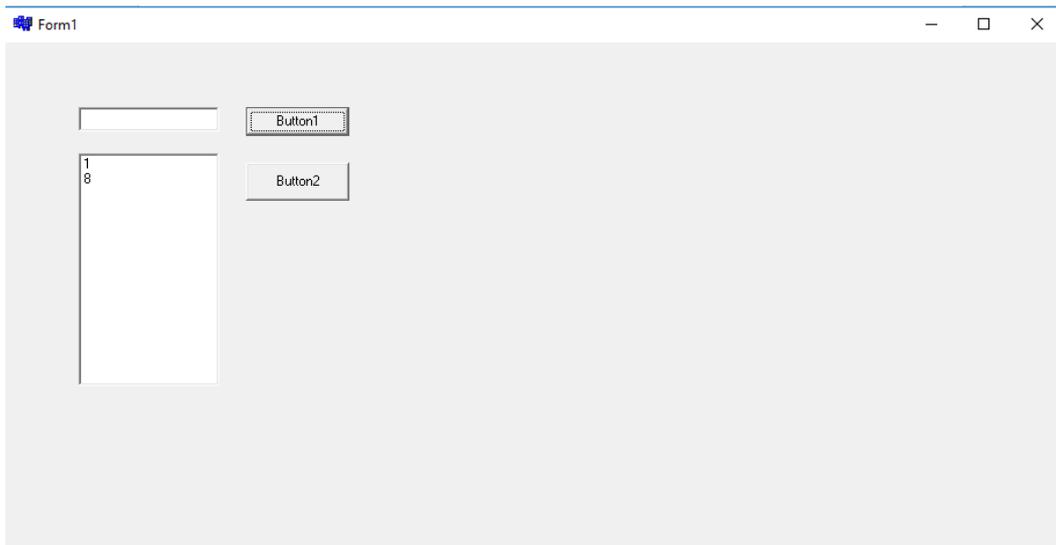
A. Screenshot Program

1. Piring/kertas ditumpuk

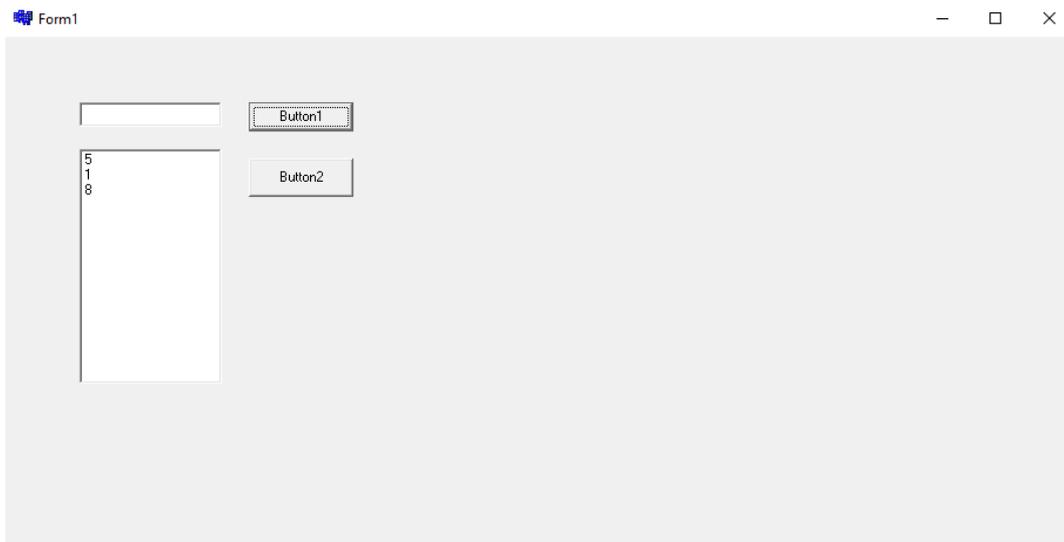
Pada proses penumpukan piring/kertas, data yang diinputkan akan tampil pada listbox dengan posisi data pertama yang diinputkan berada pada posisi paling bawah dan data terakhir masuk berada pada posisi paling atas pada listbox.



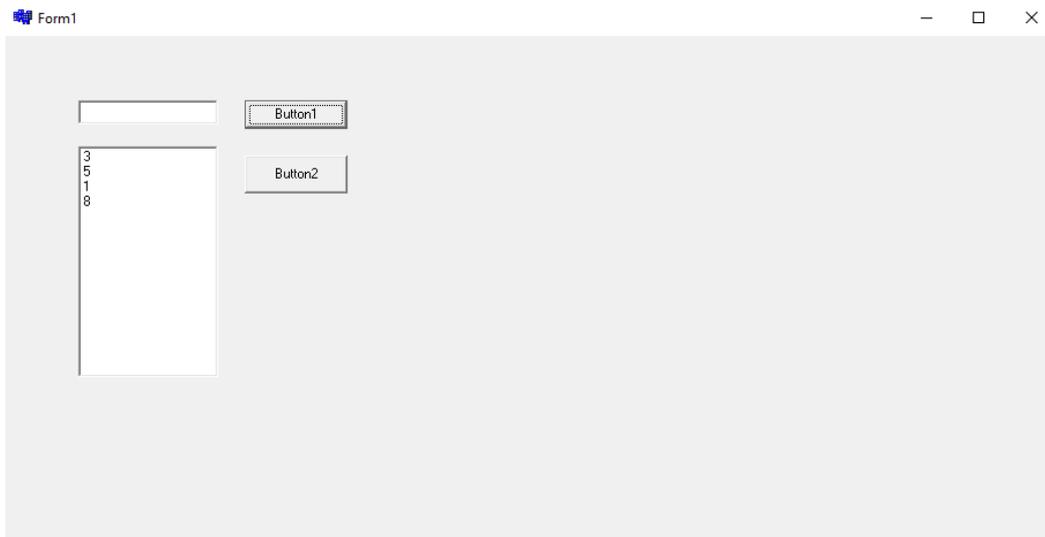
Gambar 1. Angka 8 diinputkan sebagai data pertama



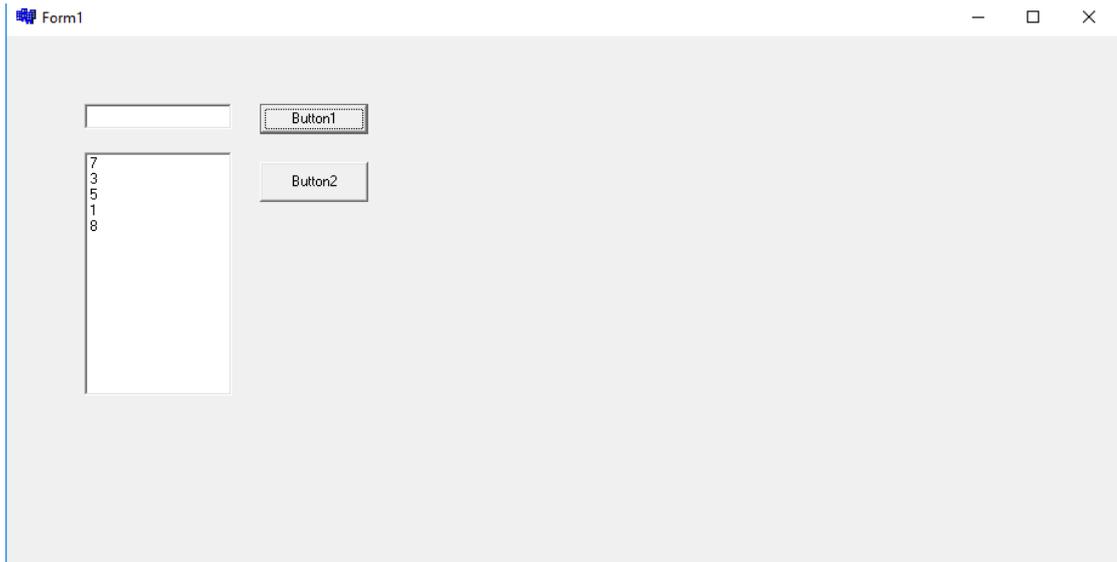
Gambar 2. Angka 1 diinputkan sebagai data kedua



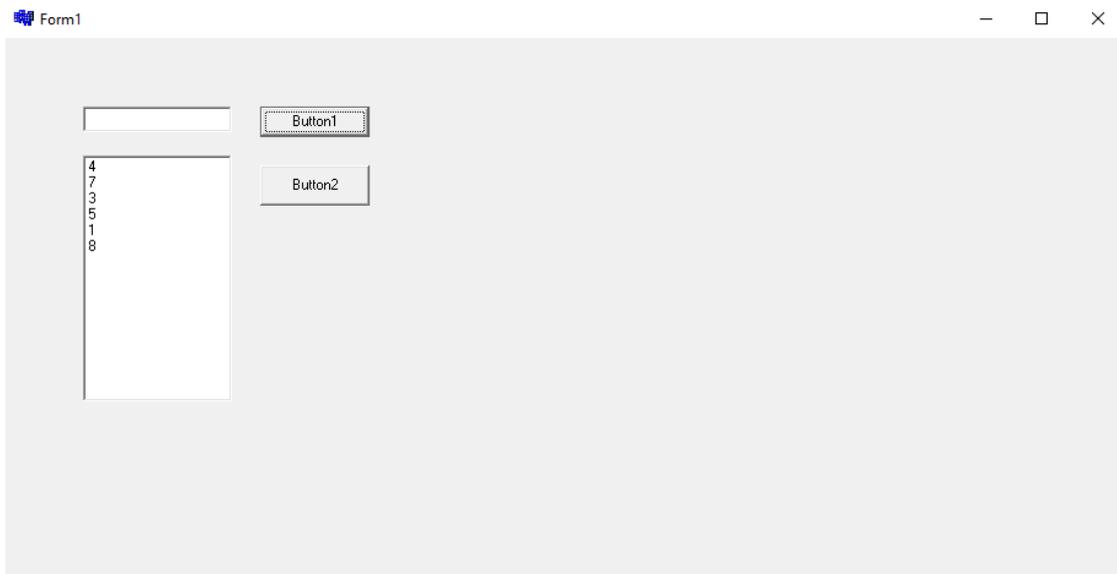
Gambar 3. Selanjutnya angka 5 diinputkan



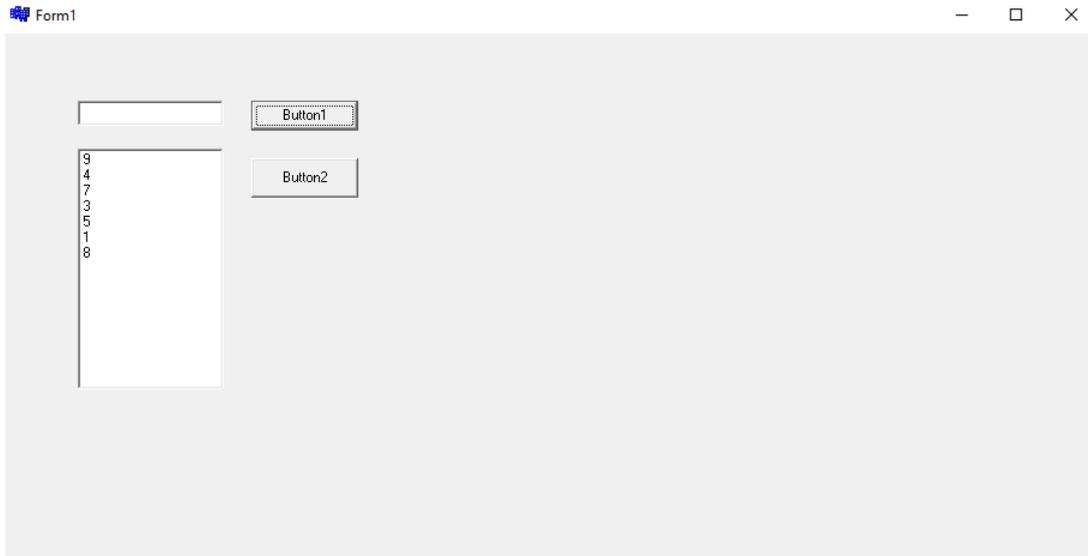
Gambar 4. Inputan Selanjutnya angka 3



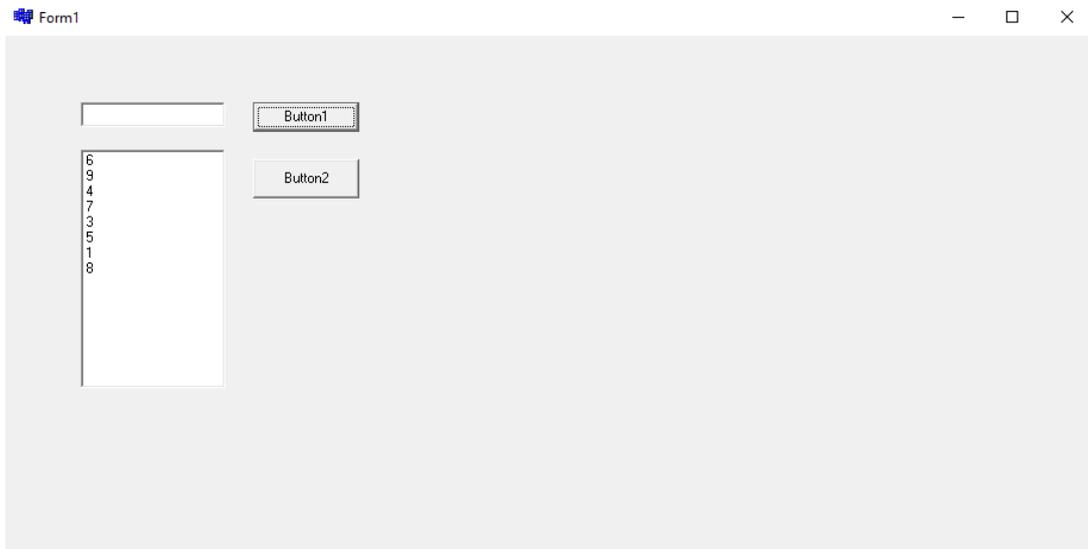
Gambar 5. Inputan Selanjutnya angka 7



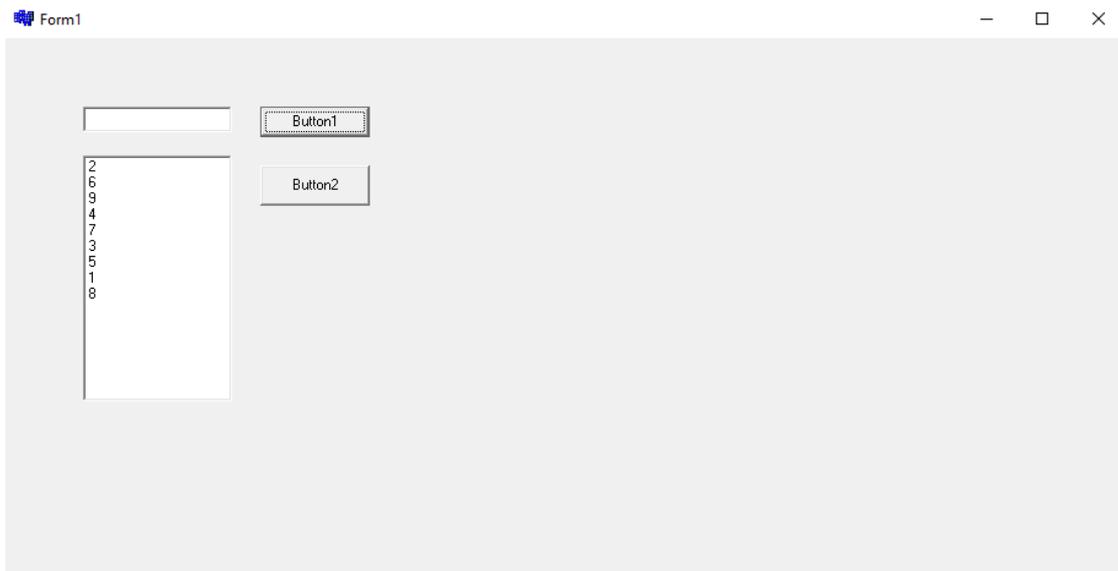
Gambar 6. Inputan Selanjutnya angka 4



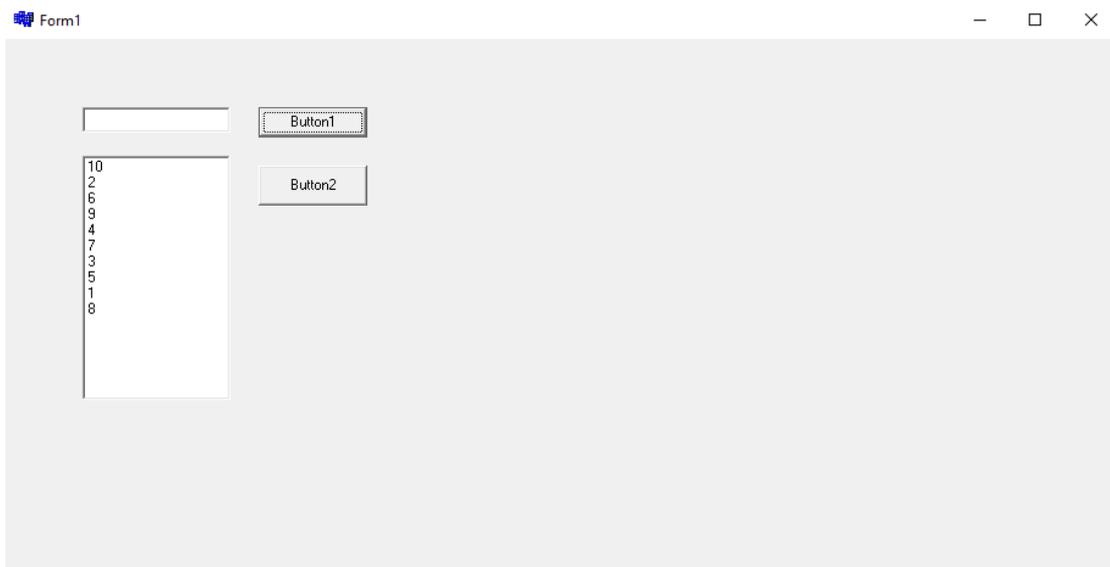
Gambar 7. Inputan Selanjutnya angka 9



Gambar 8. Inputan Selanjutnya angka 6



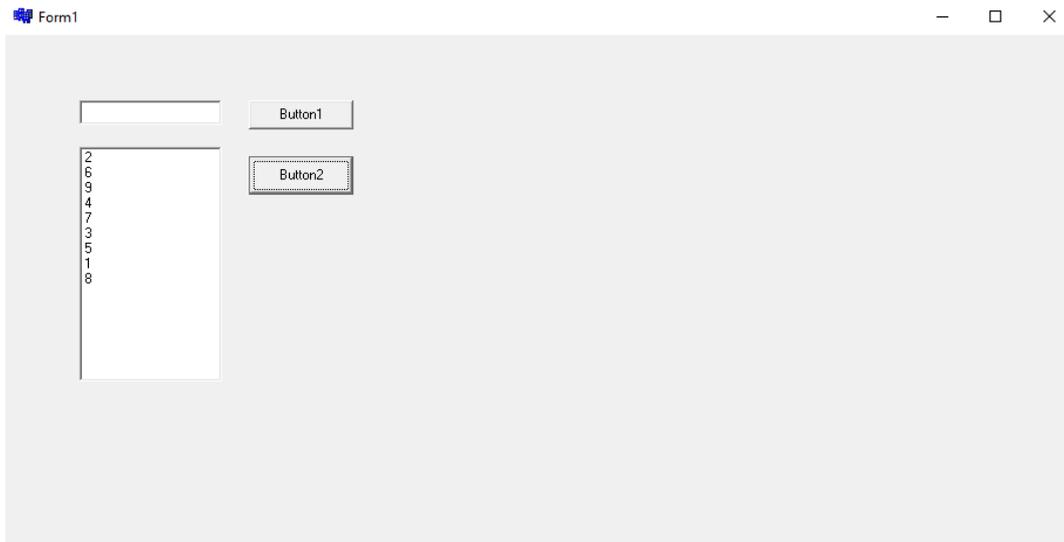
Gambar 9. Inputan Selanjutnya angka 2



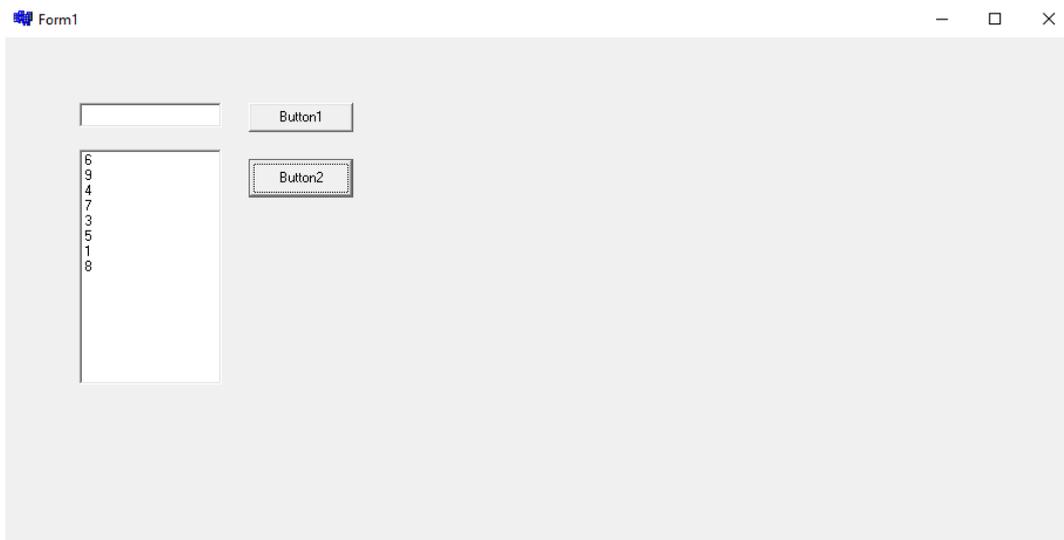
Gambar 10. Inputan Selanjutnya angka 10

2. Pengambilan tumpukan

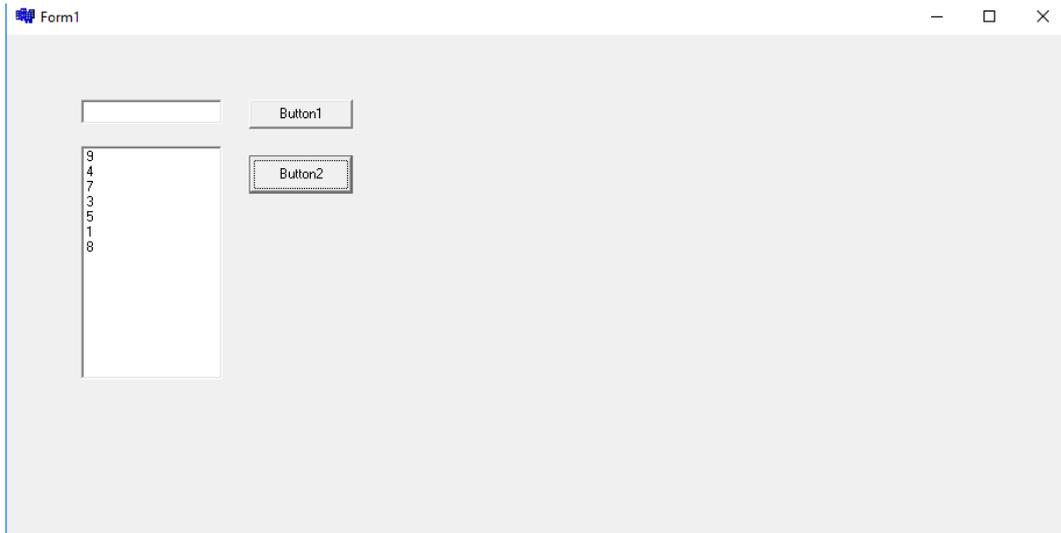
Pada saat button2 ditekan, maka data yg paling atas akan hilang dan data akan bergeser satu per satu ke atas. Lihat gambar 11.



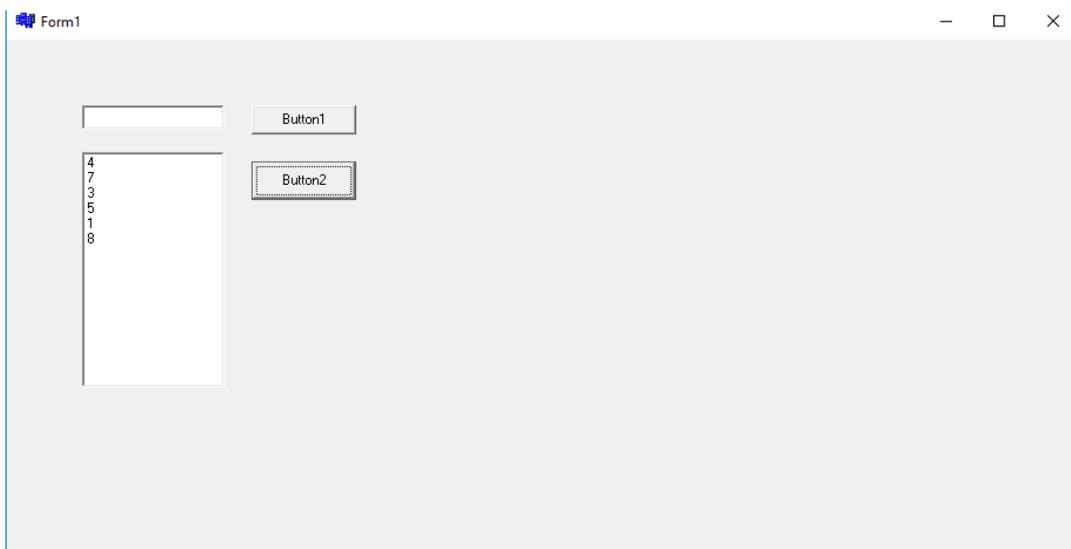
Gambar 11. Pengambilan tumpukan ke-1, angka 10 keluar dan data akan bergeser satu per satu ke atas



Gambar 12. Pengambilan tumpukan ke-2, angka 2 keluar dan data akan bergeser satu per satu ke atas

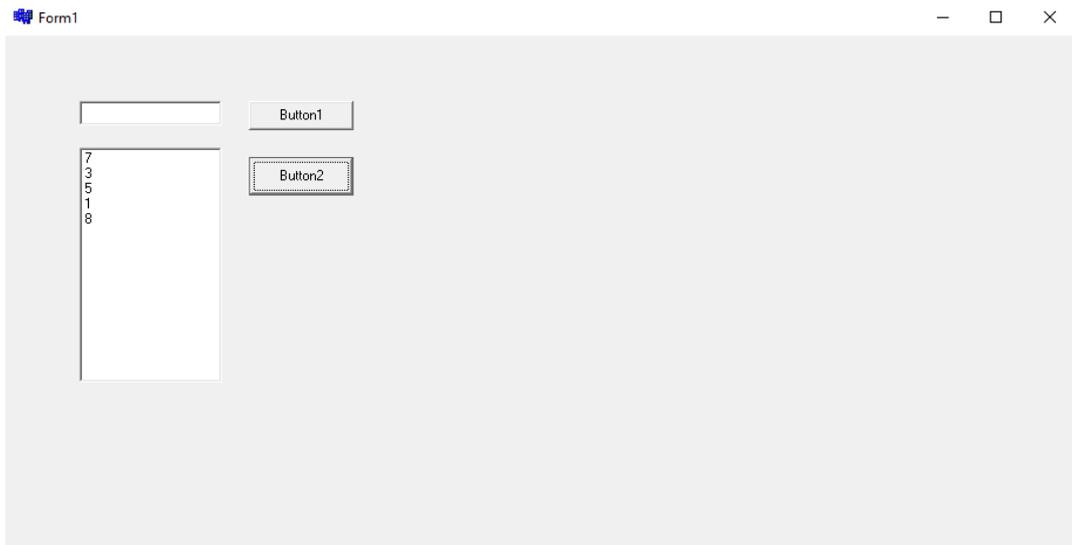


Gambar 13. Pengambilan tumpukan ke-3, angka 6 keluar dan data akan bergeser satu per satu ke atas

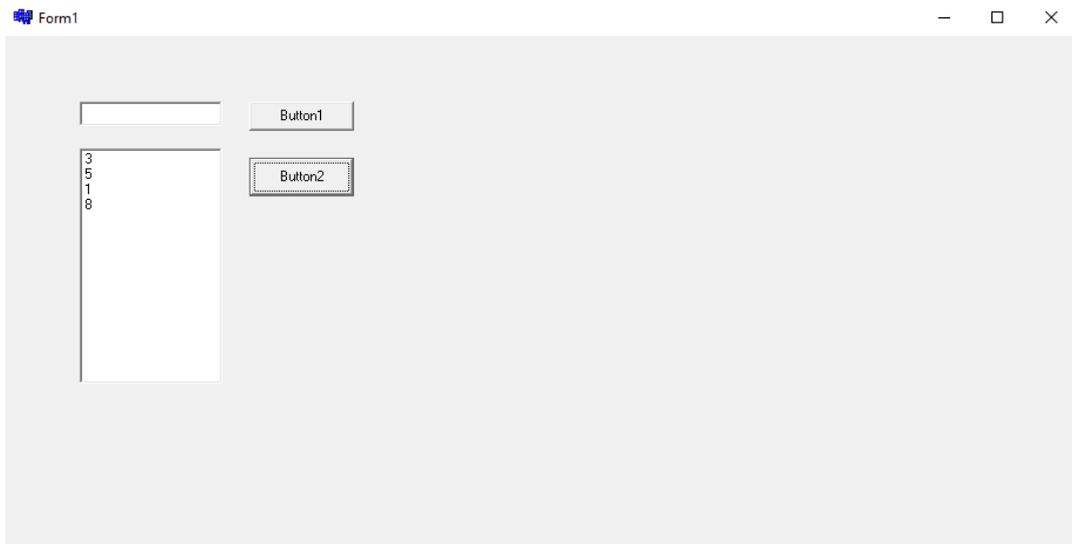


Gambar 1

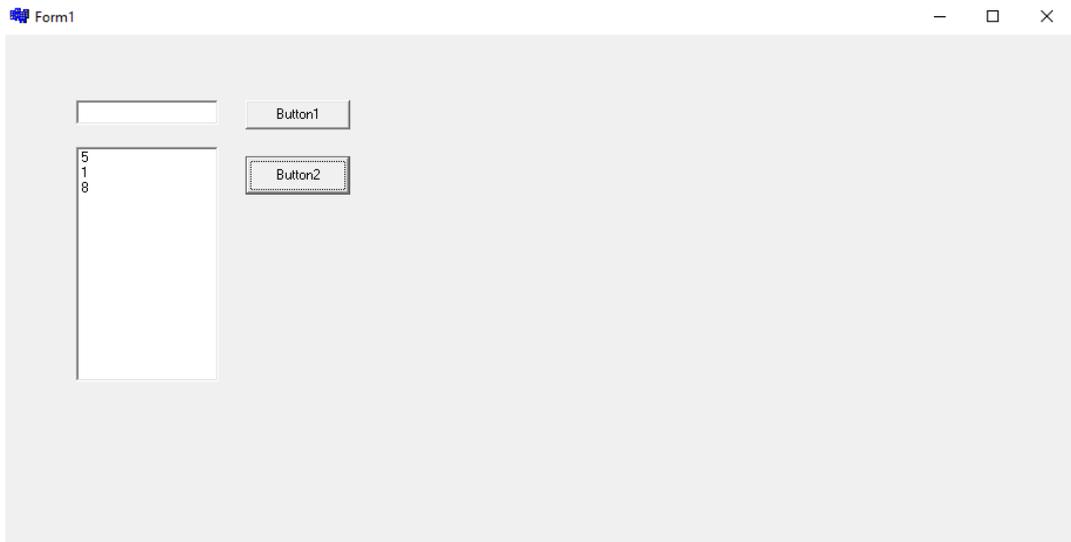
Gambar 14. Pengambilan tumpukan ke-4, angka 9 keluar dan data akan bergeser satu per satu ke atas



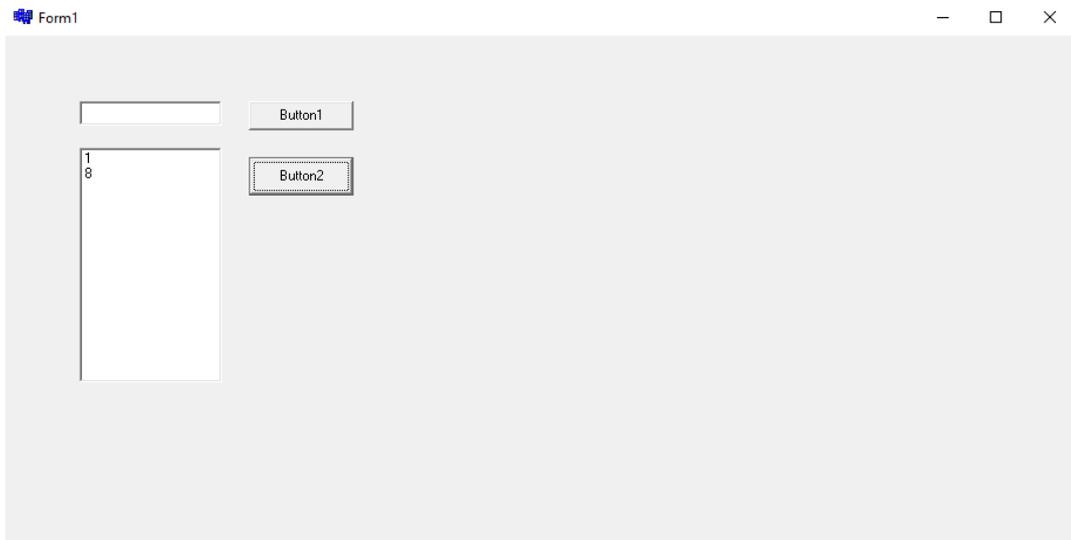
Gambar 15. Pengambilan tumpukan ke-5, angka 4 keluar dan data akan bergeser satu per satu ke atas



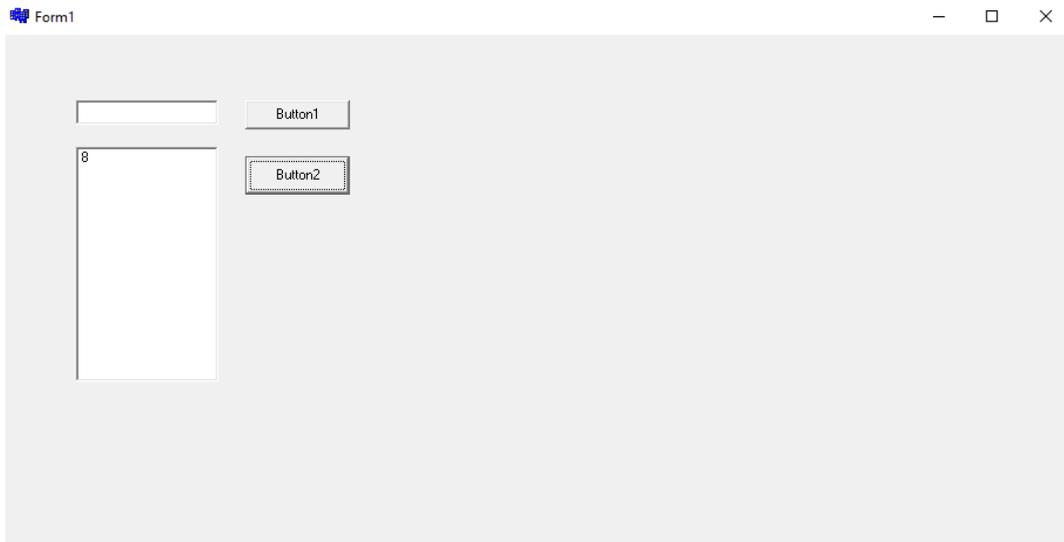
Gambar 16. Pengambilan tumpukan ke-6, angka 7 keluar dan data akan bergeser satu per satu ke atas



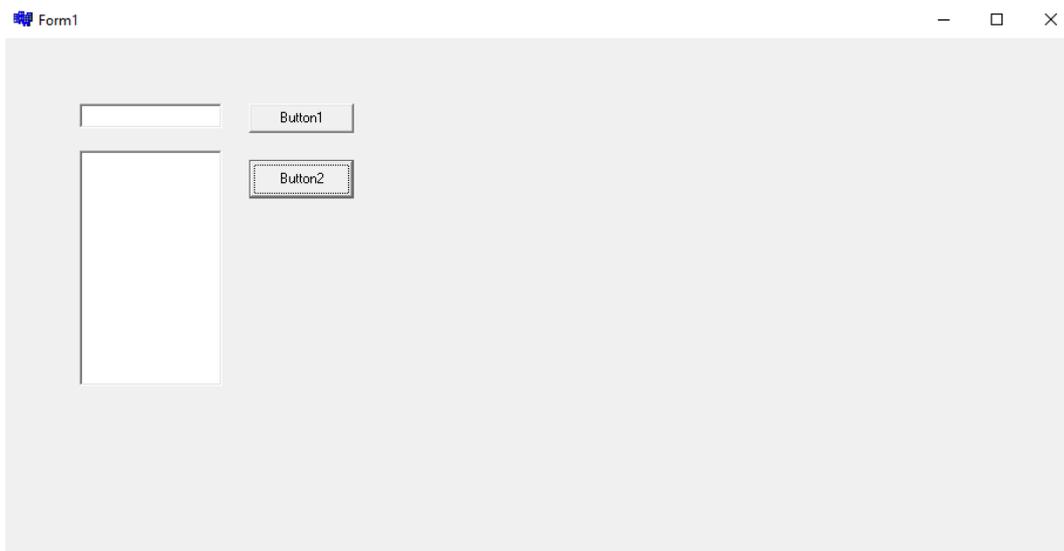
Gambar 17. Pengambilan tumpukan ke-7, angka 3 keluar dan data akan bergeser satu per satu ke atas



Gambar 18. Pengambilan tumpukan ke-8, angka 5 keluar dan data akan bergeser satu per satu ke atas



Gambar 19. Pengambilan tumpukan ke-9, angka 1 keluar dan angka 8 akan bergeser ke atas



Gambar 20. Pengambilan tumpukan ke-10, angka 8 keluar dan tumpukan menjadi kosong