



Memento Pattern

Pola Desain Perangkat Lunak
Tjatur Kandaga G.
Fakultas IT - UK Maranatha

Mengenai Memento Pattern

- Memento pattern termasuk dalam golongan behavioural pattern.
- Memento memungkinkan untuk meng-capture dan meng-eksternalisasi internal state dari objek sehingga dapat di-restore kembali. Semuanya dilakukan tanpa melanggar enkapsulasi.



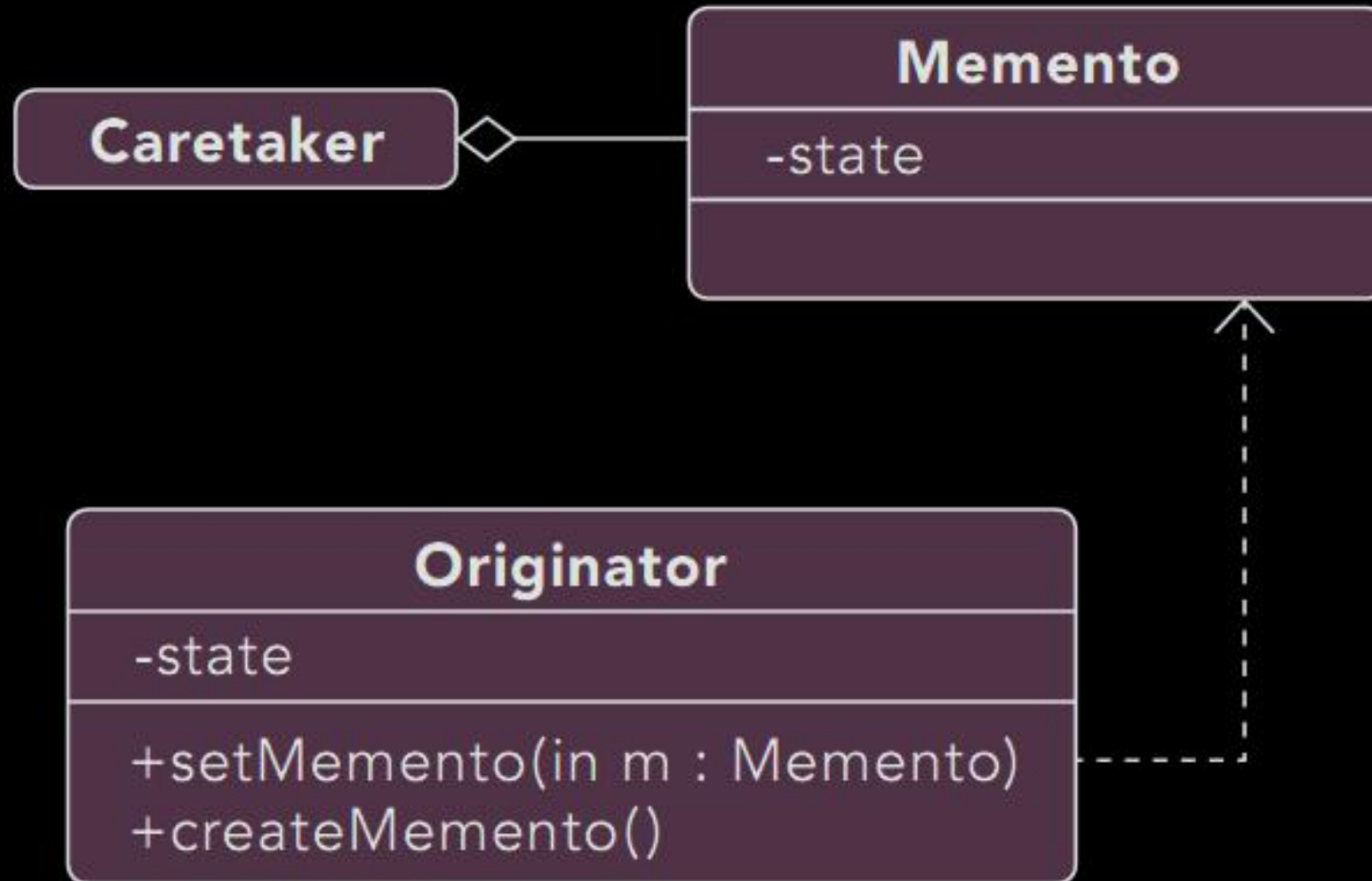
Mengenal Memento Pattern

- Memento pattern menurut buku GoF: *Without violating encapsulation, capture and externalize an object's internal state so that the object can be restored to this state later.*



Mengenal Memento Pattern

- Diagram kelas memento pattern:



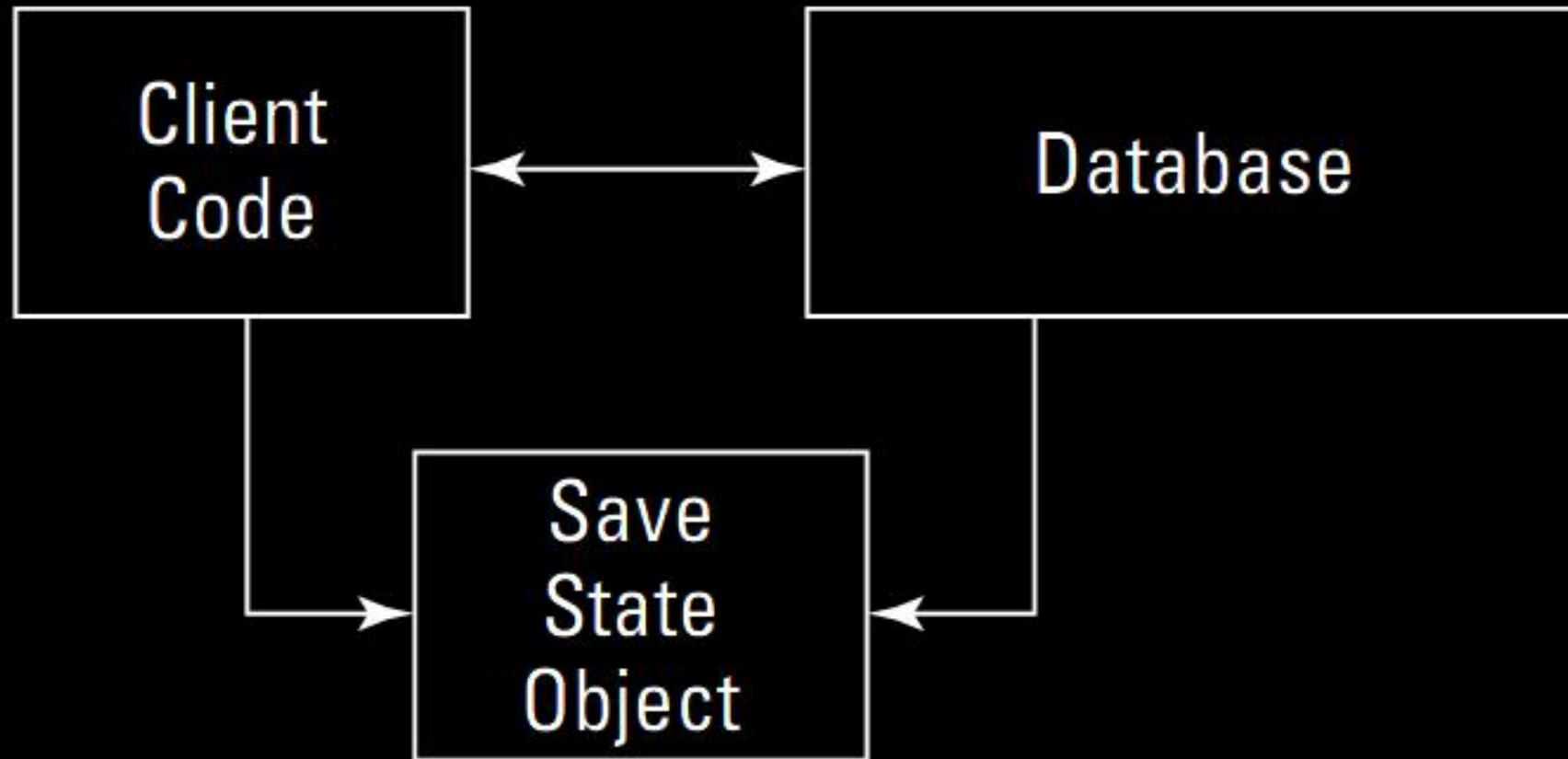
Mengenal Memento Pattern

- Dengan memento pattern kita mendefinisikan sebuah kelas yang dapat mengingat internal state dari objek yang lain.
- Caretaker mengontrol pembuatan Memento.
- Originator akan menggunakan Memento ketika me-restore state-nya.



Mengenal Memento Pattern

- Melakukan save state tanpa memento pattern akan melanggar encapsulation:



Mengenal Memento Pattern

- Melakukan save state dengan memento pattern:



Mengenal Memento Pattern

- Memento pattern digunakan ketika:
 - State internal dari objek perlu disimpan supaya dapat di-restore lagi.
 - Internal state tidak dapat di-ekspose oleh interface tanpa meng-ekspose implementasinya juga.
 - Enkapsulasi harus tetap dipertahankan.



Contoh Memento Pattern

- Contoh Memento:
 - Fitur undo dapat diimplementasikan dengan menggunakan memento pattern. Undo dapat dilakukan dengan men-serialisasi state dari objek sebelum terjadinya perubahan. Jika ingin melakukan pembatalan, maka dapat dilakukan dengan men-deserialisasi state dari objek yang sudah disimpan sebelumnya.



Contoh Memento Pattern

```
1 package coba;
2
3 import java.io.Serializable;
4
5 interface Address extends Serializable {
6     public static final String EOL_STRING = System
7         .getProperty("line.separator");
8     public static final String SPACE = " ";
9     public static final String COMMA = ",";
10
11     public String getType();
12     public String getDescription();
13     public String getStreet();
14     public String getCity();
15     public String getState();
16     public String getZipCode();
17     public void setType(String newType);
18     public void setDescription(String newDescription);
19     public void setStreet(String newStreet);
20     public void setCity(String newCity);
21     public void setState(String newState);
22     public void setZipCode(String newZip);
23 }
```



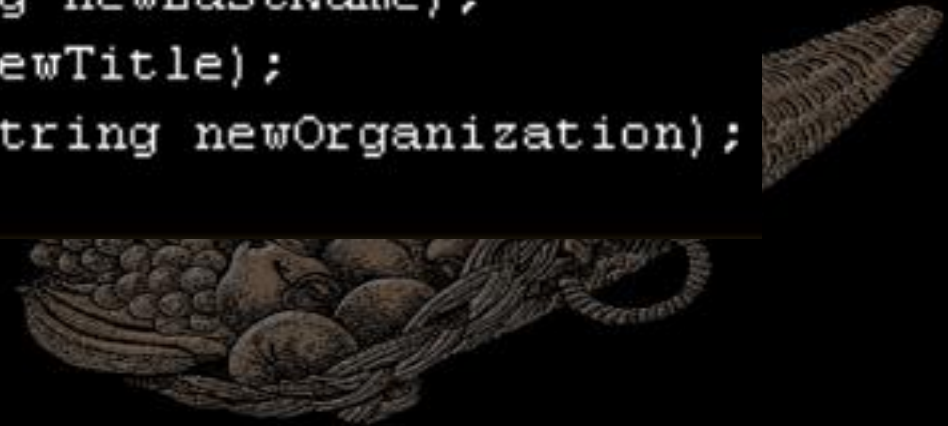
Contoh Memento Pattern

```
1 package coba;
2
3 public class AddressImpl implements Address {
4     private static final long serialVersionUID = 1L;
5     private String type;
6     private String description;
7     private String street;
8     private String city;
9     private String state;
10    private String zipCode;
11
12    public AddressImpl() {}
13
14    public AddressImpl(String newDescription, String newStreet,
15        String newCity, String newState, String newZipCode) {
16        description = newDescription;
17        street = newStreet;
18        city = newCity;
19        state = newState;
20        zipCode = newZipCode;
21    }
22
23    public String toString() {
24        return street + EOL_STRING + city + COMMA + SPACE + state
25            + SPACE + zipCode + EOL_STRING;
26    }
27
28    // insert getters & setters here
```



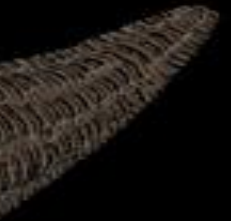
Contoh Memento Pattern

```
1 package coba;
2 import java.io.Serializable;
3
4 interface Contact extends Serializable {
5     public static final String SPACE = " ";
6     public String getFirstName();
7     public String getLastName();
8     public String getTitle();
9     public String getOrganization();
10    public void setFirstName(String newFirstName);
11    public void setLastName(String newLastName);
12    public void setTitle(String newTitle);
13    public void setOrganization(String newOrganization);
14 }
```



Contoh Memento Pattern

```
1 package coba;
2
3 public class ContactImpl implements Contact {
4     private static final long serialVersionUID = 1L;
5     private String firstName;
6     private String lastName;
7     private String title;
8     private String organization;
9     private Address address;
10
11     public ContactImpl() {
12     }
13
14     public ContactImpl(String newFirstName, String newLastName,
15         String newTitle, String newOrganization, Address newAddress) {
16         firstName = newFirstName;
17         lastName = newLastName;
18         title = newTitle;
19         organization = newOrganization;
20         address = newAddress;
21     }
22
23     public String toString() {
24         return firstName + " " + lastName;
25     }
26
27     //insert getters & setters here
28 }
```



Contoh Memento Pattern

```
1 package coba;
2
3 import java.util.ArrayList;
4
5 public class AddressBook {
6     private ArrayList<Contact> contacts = new ArrayList<Contact>();
7
8     public Object getMemento() {
9         return new AddressBookMemento(contacts);
10    }
11
12    public void setMemento(Object object) {
13        if (object instanceof AddressBookMemento) {
14            AddressBookMemento memento = (AddressBookMemento) object;
15            contacts = memento.state;
16        }
17    }
18
19    private class AddressBookMemento {
20        private ArrayList<Contact> state;
21
22        private AddressBookMemento(ArrayList<Contact> contacts) {
23            this.state = contacts;
24        }
25    }
26
```



Contoh Memento Pattern

```
27 public AddressBook() {  
28     }  
29  
30 public AddressBook(ArrayList<Contact> newContacts) {  
31     contacts = newContacts;  
32 }  
33  
34 public void addContact(Contact contact) {  
35     if (!contacts.contains(contact)) {  
36         contacts.add(contact);  
37     }  
38 }  
39  
40 public void removeContact(Contact contact) {  
41     contacts.remove(contact);  
42 }  
43  
44 public void removeAllContacts() {  
45     contacts = new ArrayList<Contact>();  
46 }  
47  
48 public ArrayList<Contact> getContacts() {  
49     return contacts;  
50 }  
51  
52 public String toString() {  
53     return contacts.toString();  
54 }  
55 }
```



Contoh Memento Pattern

```
1 package coba;
2
3 public class RunMementoPattern {
4     public static void main(String[] arguments) {
5         System.out.println("Contoh Memento pattern");
6         System.out.println();
7
8         System.out.println("Buat AddressBook");
9         AddressBook book = new AddressBook();
10
11         System.out.println("Tambahkan Contact untuk AddressBook");
12         book.addContact(new ContactImpl("Sammy", "David", "Mahasiswa",
13             "IT Maranatha", new AddressImpl()));
14         book.addContact(new ContactImpl("Annisa", "Novatia", "Assisten",
15             "FE Maranatha", new AddressImpl()));
16         book.addContact(new ContactImpl("Yosia", "Teddy", "Staff Lab",
17             "Elektro Maranatha", new AddressImpl()));
18         book.addContact(new ContactImpl("Reza", "Sanjaya", "Admin Lab",
19             "Kedokteran Maranatha", new AddressImpl()));
20
21         System.out.println("Daftar Contact:");
22         System.out.println(book);
23         System.out.println();
24     }
```

Contoh Memento Pattern

```
25      System.out.println("Buat Memento untuk address book");
26      Object memento = book.getMemento();
27      System.out.println("Memento yg terbentuk bisa dipakai untuk");
28      System.out.println(" mengeset state dari adressbook ini / yg lain");
29      System.out.println();
30
31      System.out.println("Buat data baru AddressBook");
32      book.removeAllContacts();
33      book.addContact(new ContactImpl("Himawan", "Setiawan", "Mahasiswa",
34                                   "Sipil UKM", new AddressImpl()));
35      book.addContact(new ContactImpl("Ivana", "Widjaya", "Mahasiswa",
36                                   "Psikologi UKM", new AddressImpl()));
37      book.addContact(new ContactImpl("Yurio", "Tanwiwin", "Assiten",
38                                   "Teknik Industri UKM", new AddressImpl()));
39      book.addContact(new ContactImpl("Antony", "Senjaya", "Mahasiswa",
40                                   "Magister Manajemen ", new AddressImpl()));
41
42      System.out.println("Daftar Contacts:");
43      System.out.println(book);
44      System.out.println();
45      System.out.println("Gunakan Memento object untuk mengembalikan");
46      System.out.println(" AddressBook ke state sebelumnya.");
47      book.setMemento(memento);
48      System.out.println("AddressBook direstore. Data Contact:");
49      System.out.println(book);
50  }
51 }
```

Contoh Memento Pattern

Contoh Memento pattern

Buat AddressBook

Tambahkan Contact untuk AddressBook

Daftar Contact:

[Sammy David, Annisa Novatia, Yosia Teddy, Reza Sanjaya]

Buat Memento untuk address book

Memento yg terbentuk bisa dipakai untuk
mengeset state dari adressbook ini / yg lain

Buat data baru AddressBook

Daftar Contacts:

[Himawan Setiawan, Ivana Widjaya, Yurio Tanwiwin, Antony Sanjaya]

Gunakan Memento object untuk mengembalikan
AddressBook ke state sebelumnya.

AddressBook direstore. Data Contact:

[Sammy David, Annisa Novatia, Yosia Teddy, Reza Sanjaya]



Contoh Memento (C#)

```
1  using System;
2      using System.Collections.Generic;
3      using System.Linq;
4      using System.Text;
5
6  namespace MementoExample01
7  {
8      public interface iCopyable
9      {
10         Object Copy();
11     }
12 }
```



Contoh Memento (C#)

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5
6 namespace MementoExample01
7 {
8     public class Address:ICopyable
9     {
10         private string _type;
11         private string _street;
12         private string _city;
13         private string _state;
14         private string _zipcode;
15
16         public const string EOL_STRING = "\n";
17         public const string COMA = ", ";
18         public const string HOME = "home";
19         public const string WORK = "work";
20
21         public object Copy()
22         {
23             return new Address(_street, _city, _state, _zipcode);
24         }
25
26         public Address()
27         {
28
29         }
30     }
```



Contoh Memento (C#)

```
31 public Address(string initType, string initStreet, string initCity,  
32     string InitState, string initZip)  
33 {  
34     _type = initType;  
35     _street = initStreet;  
36     _city = initCity;  
37     _state = InitState;  
38     _zipcode = initZip;  
39 }  
40  
41 public Address(string initStreet, string initCity, string InitState,  
42     string initZip): this(WORK, initStreet, initCity, InitState, initZip) { }  
43  
44 public Address(string initType)  
45 {  
46     _type = initType;  
47 }  
48  
49 public override string ToString()  
50 {  
51     return "\t" + _street + COMA + " " + EOL_STRING + "\t" + _city + COMA +  
52         " " + _state + " " + _zipcode;  
53 }  
54 }  
55 }
```

Contoh Memento (C#)

```
1  using System;
2      using System.Collections.Generic;
3      using System.Linq;
4      using System.Text;
5
6  namespace MementoExample01
7  {
8      public interface iContact
9      {
10         string FirstName {get; set;}
11         string LastName { get; set; }
12         string Title { get; set; }
13         string Organization { get; set; }
14     }
15 }
```



Contoh Memento (C#)

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5
6 namespace MementoExample01
7 {
8     class ContactImplement:iContact
9     {
10         private const long serialVersionUID = 1L;
11         private String _firstName;
12         private String _lastName;
13         private String _title;
14         private String _organization;
15         private Address _address;
16
17
18         public ContactImplement(String newFirstName, String newLastName,
19             String newTitle, String newOrganization, Address newAddress)
20         {
21             _firstName = newFirstName;
22             _lastName = newLastName;
23             _title = newTitle;
24             _organization = newOrganization;
25             _address = newAddress;
26         }
27
28         public override string ToString()
29         {
30             return _firstName + " " + _lastName;
31         }
32     }
```



Contoh Memento (C#)

```
33     public String FirstName
34     {
35         get { return _firstName; }
36         set { _firstName = value; }
37     }
38
39     public String LastName
40     {
41         get { return _lastName; }
42         set { _lastName = value; }
43     }
44
45     public String Title
46     {
47         get { return _title; }
48         set { _title = value; }
49     }
50
51     public String Organization
52     {
53         get { return _organization; }
54         set { _organization = value; }
55     }
56
57     public Address Address
58     {
59         get { return _address; }
60         set { _address = value; }
61     }
62 }
63 }
```



Contoh Memento (C#)

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5
6 namespace MementoExample01
7 {
8     public class Addressbook
9     {
10         private List<iContact> contacts = new List<iContact>();
11
12         public object getMemento()
13         {
14             return new AddressbookMemento(contacts);
15         }
16
17         public void setMemento(Object o)
18         {
19             if (o is AddressbookMemento)
20             {
21                 AddressbookMemento memento = (AddressbookMemento) o;
22                 contacts = memento.State;
23             }
24         }
25     }
```



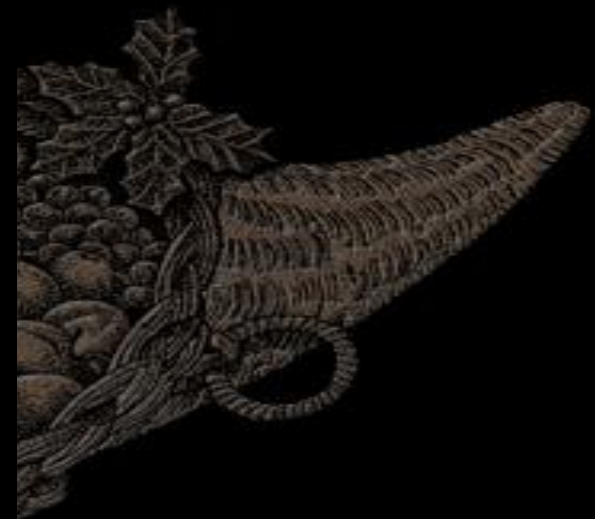
Contoh Memento (C#)

```
26 private class AddressbookMemento
27 {
28     private List<iContact> _state;
29
30     public List<iContact> State
31     {
32         get { return _state; }
33     }
34
35     public AddressbookMemento(List<iContact> contacts)
36     {
37         this._state = contacts;
38     }
39 }
40
41 public Addressbook()
42 {
43
44 }
45
46 public Addressbook(List<iContact> newContacts)
47 {
48     contacts = newContacts;
49 }
50
51 public void addContact(iContact contact)
52 {
53     if (!contacts.Contains(contact))
54     {
55         contacts.Add(contact);
56     }
57 }
58
```



Contoh Memento (C#)

```
59 public void RemoveContact(iContact contact)
60 {
61     contacts.Remove(contact);
62 }
63
64 public void RemoveAllContacts()
65 {
66     contacts = new List<iContact>();
67 }
68
69 public List<iContact> getContacts()
70 {
71     return contacts;
72 }
73
74 public override string ToString()
75 {
76     string temp="";
77     foreach (var item in contacts)
78     {
79         temp += item.ToString() + " - ";
80     }
81     return temp;
82 }
83 }
84 }
```

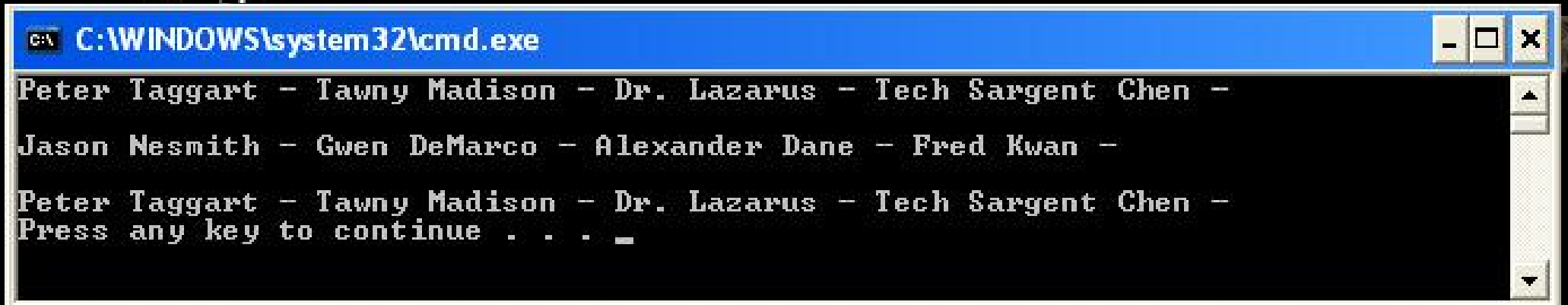


Contoh Memento (C#)

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5
6 namespace MementoExample01
7 {
8     class Program
9     {
10         static void Main(string[] args)
11         {
12             Addressbook book = new Addressbook();
13             book.addContact(new ContactImplement(
14                 "Peter", "Taggart", "Commander", "NSEA Protector", new Address()));
15             book.addContact(new ContactImplement(
16                 "Tawny", "Madison", "Lieutenant", "NSEA Protector", new Address()));
17             book.addContact(new ContactImplement("Dr.", "Lazarus", "Dr.",
18                 "NSEA Protector", new Address()));
19             book.addContact(new ContactImplement(
20                 "Tech Sargent", "Chen", "Tech Sargent", "NSEA Protector", new Address()));
21
22             Console.WriteLine(book);
23             Console.WriteLine();
24             object memento = book.getMemento();
25
```


Contoh Memento (C#)

```
26         book.RemoveAllContacts();
27
28         book.AddContact(new ContactImplement(
29             "Jason", "Nesmith", "", "Actor's Guild", new Address()));
30
31         book.AddContact(new ContactImplement(
32             "Gwen", "DeMarco", "", "Actor's Guild", new Address()));
33
34         book.AddContact(new ContactImplement(
35             "Alexander", "Dane", "", "Actor's Guild", new Address()));
36
37         book.AddContact(new ContactImplement(
38             "Fred", "Kwan", "", "Actor's Guild", new Address()));
39
40         Console.WriteLine(book);
41         Console.WriteLine();
42         book.SetMemento(memento);
43         Console.WriteLine(book);
44     }
45 }
46 }
```



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The output of the program is displayed in a monospaced font. It shows a list of contacts: "Peter Taggart - Tawny Madison - Dr. Lazarus - Tech Sargent Chen -", followed by "Jason Nesmith - Gwen DeMarco - Alexander Dane - Fred Kwan -", and then the same list of contacts again. The prompt "Press any key to continue . . . _" is visible at the bottom.

```
C:\WINDOWS\system32\cmd.exe
Peter Taggart - Tawny Madison - Dr. Lazarus - Tech Sargent Chen -
Jason Nesmith - Gwen DeMarco - Alexander Dane - Fred Kwan -
Peter Taggart - Tawny Madison - Dr. Lazarus - Tech Sargent Chen -
Press any key to continue . . . _
```

Memento Pattern saving an object data in
it's internal class object

