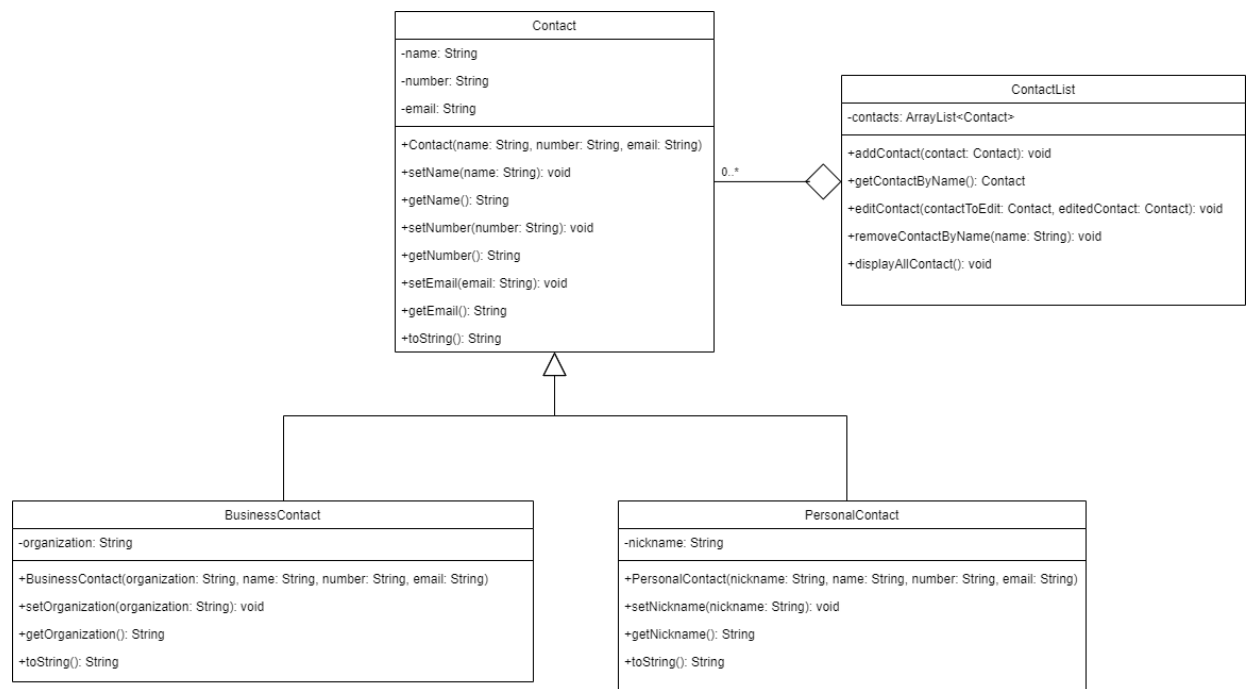


Laporan Praktikum Minggu 4 (Inheritance)

Nama Kelompok:

1. Abiem Akmal Fadhil - 11231002
2. Hari Yatun Nasifa - 11231030
3. Maulida Rahmayanti - 11231038
4. **Muhammad Iqbal Alexandre Saputra – 11231056**

Class Diagram Manajemen Kontak



Source Code

No.	org/tugas/tugas4/Contacts/Contact.java
1.	package org.tugas.tugas4.Contacts;
2.	
3.	public class Contact {
4.	private String name, number, email;
5.	
6.	public Contact(String name, String number, String email) {
7.	this.name = name;
8.	this.number = number;
9.	this.email = email;
10.	}
11.	
12.	public String getName() {
13.	return name;
14.	}
15.	
16.	public void setName(String name) {
17.	this.name = name;
18.	}
19.	
20.	public String getNumber() {
21.	return number;
22.	}
23.	
24.	public void setNumber(String number) {
25.	this.number = number;
26.	}
27.	
28.	public String getEmail() {
29.	return email;
30.	}
31.	
32.	public void setEmail(String email) {
33.	this.email = email;
34.	}
35.	
36.	public String toString() {
37.	return "Name: " + this.name + "\nPhone Number: " + this.number + "\nEmail: " + this.email;
38.	}
39.	}

No.	org/tugas/tugas4/Contacts/PersonalContact.java
1.	package org.tugas.tugas4.Contacts;
2.	
3.	public class PersonalContact extends Contact {
4.	private String nickname;
5.	
6.	public PersonalContact(String name, String number, String email, String nickname) {
7.	super(name, number, email);
8.	this.nickname = nickname;
9.	}
10.	
11.	public String getNickname() {
12.	return nickname;
13.	}
14.	
15.	public void setNickname(String nickname) {
16.	this.nickname = nickname;
17.	}
18.	
19.	@Override
20.	public String toString() {
21.	return super.toString() + "\nNickname: " + this.nickname;
22.	}
23.	
24.	}
25.	

No.	org/tugas/tugas4/Contacts/BusinessContact.java
1.	package org.tugas.tugas4.Contacts;
2.	
3.	public class BusinessContact extends Contact {
4.	private String organization;
5.	
6.	public BusinessContact(String name, String number, String email, String organization) {
7.	super(name, number, email);
8.	this.organization = organization;
9.	}
10.	
11.	public String getOrganization() {
12.	return organization;
13.	}
14.	
15.	public void setOrganization(String organization) {
16.	this.organization = organization;
17.	}
18.	
19.	@Override
20.	public String toString() {
21.	return super.toString() + "\nOrganization: " + this.organization;
22.	}
23.	
24.	}
25.	

No	org/tugas/tugas3/ContactList.java
1.	package org.tugas.tugas4;
2.	
3.	import java.util.ArrayList;
4.	import org.tugas.tugas4.Contacts.*;
5.	
6.	public class ContactList {
7.	private ArrayList<Contact> contacts = new ArrayList<>();
8.	
9.	public void addContact(Contact contact) {
10.	this.contacts.add(contact);
11.	}
12.	
13.	public void editContact(Contact contactToEdit, Contact editedContact) {
14.	for (Contact contact : this.contacts) {
15.	if (contactToEdit.equals(contact)) {
16.	contact.setName(editedContact.getName());
17.	contact.setNumber(editedContact.getNumber());
18.	contact.setEmail(editedContact.getEmail());
19.	
20.	if (contactToEdit instanceof BusinessContact) {
21.	BusinessContact editedBusinessContact = (BusinessContact)
22.	editedContact;
23.	BusinessContact businessContact = (BusinessContact) contact;
24.	businessContact.setOrganization(editedBusinessContact.getOrganization());
25.	} else if (contactToEdit instanceof PersonalContact) {
26.	PersonalContact editedPersonalContact = (PersonalContact)
27.	editedContact;
28.	PersonalContact personalContact = (PersonalContact) contact;
29.	personalContact.setNickname(editedPersonalContact.getNickname());
30.	}
31.	}
32.	}
33.	}
34.	
35.	public void removeContactByName(String name) {
36.	for (Contact contact : this.contacts) {
37.	if (name.equals(contact.getName())) {
38.	this.contacts.remove(contact);
39.	break;
40.	}
41.	}
42.	}
43.	
44.	public Contact getContactByName(String name) {
45.	for (Contact contact : this.contacts) {
46.	if (contact.getName().equals(name)) {
47.	return contact;
48.	}
49.	}
50.	return null;
51.	}
52.	
53.	public void displayAllContact() {
54.	if (this.contacts.size() > 0) {
55.	int counter = 1;
56.	
57.	for (Contact contact : this.contacts) {

58.	String contactName = counter + ". " + contact.getName();
59.	if (contact instanceof PersonalContact) {
60.	contactName += " - Personal";
61.	} else if (contact instanceof BusinessContact) {
62.	contactName += " - Business";
63.	}
64.	System.out.println(contactName);
65.	counter++;
66.	}
67.	} else {
68.	System.out.println("You have no contact.");
69.	}
70.	
71.	}
72.	}
73.	
74.	

No	org/tugas/tugas3/Main.java
1.	package org.tugas.tugas4;
2.	
3.	import java.util.Scanner;
4.	import org.tugas.tugas4.Contacts.*;
5.	
6.	public class Main {
7.	static Scanner scanner = new Scanner(System.in);
8.	
9.	public static void main() {
10.	ContactList contactList = new ContactList();
11.	
12.	while (true) {
13.	displayMenu();
14.	int choice = scanner.nextInt();
15.	scanner.nextLine();
16.	if (choice != 0) {
17.	switch (choice) {
18.	case 1:
19.	contactList.displayAllContact();
20.	System.out.print("Press enter to continue");
21.	scanner.nextLine();
22.	break;
23.	case 2:
24.	System.out.println("1. Default Contact");
25.	System.out.println("2. Personal Contact");
26.	System.out.println("3. Business Contact");
27.	System.out.print("Choose a contact type:");
28.	
29.	int contactType = scanner.nextInt();
30.	scanner.nextLine();
31.	
32.	contactList.addContact(readContact(contactType));
33.	break;
34.	case 3:
35.	System.out.print("Enter contact name: ");
36.	String oldName = scanner.nextLine();
37.	Contact contact = contactList.getContactByName(oldName);
38.	
39.	if (contact instanceof PersonalContact) {

```

40.         contactList.editContact(contact, readContact(2));
41.     } else if (contact instanceof BusinessContact) {
42.         contactList.editContact(contact, readContact(3));
43.     } else if (contact instanceof Contact) {
44.         contactList.editContact(contact, readContact(1));
45.     }
46.     break;
47.     case 4:
48.         System.out.print("Enter name: ");
49.         String name = scanner.nextLine();
50.
51.         contactList.removeContactByName(name);
52.         break;
53.     case 5:
54.         System.out.print("Enter name: ");
55.         name = scanner.nextLine();
56.
57. System.out.println(contactList.getContactByName(name).toString());
58.         System.out.print("Press enter to continue");
59.         scanner.nextLine();
60.         break;
61.     default:
62.         break;
63.     }
64.     } else {
65.         break;
66.     }
67. }
68. scanner.close();
69. }
70.
71. private static void displayMenu() {
72.     System.out.println("::: Contact Menu :::");
73.     System.out.println("1. Show all contacts");
74.     System.out.println("2. Add a contact");
75.     System.out.println("3. Edit a contact");
76.     System.out.println("4. Delete a contact");
77.     System.out.println("5. Show contact detail");
78.     System.out.println("0. Exit");
79.     System.out.print("Choose one: ");
80. }
81.
82. private static Contact readContact(int contactType) {
83.     System.out.print("Enter new name: ");
84.     String name = scanner.nextLine();
85.
86.     System.out.print("Enter new phone number: ");
87.     String phoneNum = scanner.nextLine();
88.
89.     System.out.print("Enter new email: ");
90.     String email = scanner.nextLine();
91.
92.     if (contactType == 1) {
93.         return new Contact(name, phoneNum, email);
94.     } else if (contactType == 2) {
95.         System.out.print("Enter a nickname: ");
96.         String nickname = scanner.nextLine();
97.         return new PersonalContact(name, phoneNum, email, nickname);
98.     } else if (contactType == 3) {
99.         System.out.print("Enter an organization: ");
100.        String organization = scanner.nextLine();

```

101.	return new BusinessContact(name, phoneNum, email, organization);
102.	}
103.	
104.	return null;
105.	}
106.	}

Screenshot Manajemen Kontak

```

::: Contact Menu :::
1. Show all contacts
2. Add a contact
3. Edit a contact
4. Delete a contact
5. Show contact detail
0. Exit
Choose one: █

```

```

::: Contact Menu :::
1. Show all contacts
2. Add a contact
3. Edit a contact
4. Delete a contact
5. Show contact detail
0. Exit
Choose one: 1
You have no contact.
Press enter to continue █

```

```

::: Contact Menu :::
1. Show all contacts
2. Add a contact
3. Edit a contact
4. Delete a contact
5. Show contact detail
0. Exit
Choose one: 2
1. Default Contact
2. Personal Contact
3. Business Contact
Choose a contact type:1 █

```

```
::: Contact Menu :::  
1. Show all contacts  
2. Add a contact  
3. Edit a contact  
4. Delete a contact  
5. Show contact detail  
0. Exit  
Choose one: 2  
1. Default Contact  
2. Personal Contact  
3. Business Contact  
Choose a contact type:1  
Enter new name: Muhammad Iqbal  
Enter new phone number: 08123456789  
Enter new email: iqbal@email.com
```

```
::: Contact Menu :::  
1. Show all contacts  
2. Add a contact  
3. Edit a contact  
4. Delete a contact  
5. Show contact detail  
0. Exit  
Choose one: 2  
1. Default Contact  
2. Personal Contact  
3. Business Contact  
Choose a contact type:2  
Enter new name: Muhammad Joko  
Enter new phone number: 088888888  
Enter new email: joko@email.com  
Enter a nickname: Joko
```



```
::: Contact Menu :::  
1. Show all contacts  
2. Add a contact  
3. Edit a contact  
4. Delete a contact  
5. Show contact detail  
0. Exit  
Choose one: 2  
1. Default Contact  
2. Personal Contact  
3. Business Contact  
Choose a contact type:3  
Enter new name: Pak Budi  
Enter new phone number: 077777777  
Enter new email: budi@email.com  
Enter an organization: ITK
```

```
::: Contact Menu :::  
1. Show all contacts  
2. Add a contact  
3. Edit a contact  
4. Delete a contact  
5. Show contact detail  
0. Exit  
Choose one: 3  
Enter contact name: Muhammad Iqbal  
Enter new name: Iqbal  
Enter new phone number: 08812  
Enter new email: newemail@email.com  
::: Contact Menu :::  
1. Show all contacts  
2. Add a contact  
3. Edit a contact  
4. Delete a contact  
5. Show contact detail  
0. Exit  
Choose one: 5  
Enter name: Iqbal  
Name: Iqbal  
Phone Number: 08812  
Email: newemail@email.com  
Press enter to continue
```

```
::: Contact Menu :::  
1. Show all contacts  
2. Add a contact  
3. Edit a contact  
4. Delete a contact  
5. Show contact detail  
0. Exit  
Choose one: 3  
Enter contact name: Muhammad Joko  
Enter new name: Joko Baru  
Enter new phone number: 088888  
Enter new email: jokobaru@email.com  
Enter a nickname: Joko New  
::: Contact Menu :::  
1. Show all contacts  
2. Add a contact  
3. Edit a contact  
4. Delete a contact  
5. Show contact detail  
0. Exit  
Choose one: 5  
Enter name: Joko Baru  
Name: Joko Baru  
Phone Number: 088888  
Email: jokobaru@email.com  
Nickname: Joko New  
Press enter to continue
```

```
::: Contact Menu :::  
1. Show all contacts  
2. Add a contact  
3. Edit a contact  
4. Delete a contact  
5. Show contact detail  
0. Exit  
Choose one: 3  
Enter contact name: Pak Budi  
Enter new name: Budi Baru  
Enter new phone number: 07777777  
Enter new email: budibaru@email.com  
Enter an organization: Informatika  
::: Contact Menu :::  
1. Show all contacts  
2. Add a contact  
3. Edit a contact  
4. Delete a contact  
5. Show contact detail  
0. Exit  
Choose one: 5  
Enter name: Budi Baru  
Name: Budi Baru  
Phone Number: 07777777  
Email: budibaru@email.com  
Organization: Informatika  
Press enter to continue
```

```
::: Contact Menu :::  
1. Show all contacts  
2. Add a contact  
3. Edit a contact  
4. Delete a contact  
5. Show contact detail  
0. Exit
```

Choose one: 4

Enter name: Budi Baru

```
::: Contact Menu :::  
1. Show all contacts  
2. Add a contact  
3. Edit a contact  
4. Delete a contact  
5. Show contact detail  
0. Exit
```

Choose one: 1

```
1. Iqbal  
2. Joko Baru - Personal  
Press enter to continue
```

```
::: Contact Menu :::  
1. Show all contacts  
2. Add a contact  
3. Edit a contact  
4. Delete a contact  
5. Show contact detail  
0. Exit
```

Choose one: 0

BUILD SUCCESSFUL in 10m 46s

2 actionable tasks: 1 executed, 1 up-to-date

D:\School\Kuliah\Semester 2\PBO\Praktikum\Tugas 4\tugas_gradle>

Penjelasan

Contact.java

No.	Deskripsi
1	Baris ini menyatakan tempat kelas Contact berada.
3	Membuat class Contact dengan akses public
4	Membuat variabel private name, number, email dengan tipe data String
6-10	Constructor class Contact yang menerima parameter name, number, email
12-18	Metode getter dan setter untuk variabel name
20-26	Metode getter dan setter untuk variabel number
28-34	Metode getter dan setter untuk variabel email
36-38	Metode toString untuk mencetak objek contact dengan format yang lebih mudah dibaca
9-36	Blok while menerima input operasi yang ingin pengguna lakukan dan input angka kedua untuk dihitung, dan menampilkan hasilnya secara berulang-ulang hingga pengguna memilih untuk keluar dengan cara memilih input operasi selain 1 hingga 4 dan memberhentikan while loop.
38-46	Function showMenu berfungsi menampilkan menu pilihan operasi.
47-50	Function readNumber berfungsi menerima input angka dari pengguna.
53-71	Metode metode add, subtract, multiply, divide yang menerima 2 parameter yaitu angka pertama dan angka kedua lalu mengembalikan hasilnya. Metode metode ini dipanggil setelah pengguna menginput angka kedua pada blok while.

PersonalContact.java

No.	Deskripsi
1	Baris ini menyatakan tempat kelas PersonalContact berada.
3	Membuat class PersonalContact dengan akses public yang merupakan child class dari Contact
4	Membuat variabel private nickname dengan tipe data String
6	Constructor class PersonalContact yang menerima parameter name, number, email, nickname
8	Mengisi nilai awal variabel nickname
7	Memanggil constructor dari parent class yaitu Contact untuk mengisi nilai name, number

	dan email
11-17	Metode getter dan setter untuk variabel nickname
19	Keyword @Override untuk menjelaskan bahwa method berikut adalah <i>override</i> dari method yang ada di class parent.
20-22	Metode toString untuk mencetak objek contact dengan format yang lebih mudah dibaca. Memanggil metode toString dari kelas Contact lalu menambahkan nilai variabel organization

BusinessContact.java

No.	Deskripsi
1	Baris ini menyatakan tempat kelas BusinessContact berada.
3	Membuat class BusinessContact dengan akses public yang merupakan child class dari Contact
4	Membuat variabel private organization dengan tipe data String
6	Constructor class BusinessContact yang menerima parameter name, number, email, organization
7	Memanggil constructor dari parent class yaitu Contact untuk mengisi nilai name, number dan email
8	Mengisi nilai awal variabel organization
11-17	Metode getter dan setter untuk variabel organization
19	Keyword @Override untuk menjelaskan bahwa method berikut adalah <i>override</i> dari method yang ada di class parent.
20-22	Metode toString untuk mencetak objek contact dengan format yang lebih mudah dibaca. Memanggil metode toString dari kelas Contact lalu menambahkan nilai variabel nickname

ContactList.java

No.	Deskripsi
1	Baris ini menyatakan tempat kelas ContactList berada.
3	Mengimpor kelas ArrayList dari paket java.util. Kelas ArrayList digunakan untuk menyimpan objek objek contact yang akan dibuat pengguna.
4	Mengimpor semua kelas yang ada di dalam folder Contacts
6	Membuat class ContactList.
7	Membuat objek ArrayList untuk menyimpan objek-objek contact.

9-11	Method addContact yang menerima parameter objek Contact untuk menambahkan objek contact ke ArrayList contact.
13-33	Method editContact yang menerima parameter objek Contact yang akan diedit dan objek Contact baru yang telah diedit. Program melakukan linear search untuk mencari objek Contact yang ingin diedit. Jika contact yang ingin diubah adalah class PersonalContact atau BusinessContact maka program mengubah tipe Contact sesuai dengan class awalnya lalu mengubah atribut objek contact tersebut dengan objek contact yang telah diedit menggunakan metode setter.
35-42	Method removeContactByName yang menerima parameter String name untuk menghapus objek contact dari ArrayList contact, Program melakukan linear search untuk mencari objek contact yang ingin dihapus menggunakan nama yang diterima dari parameter.
44-51	Method getContactByName yang menerima parameter String name untuk mengembalikan objek contact dari ArrayList contact berdasarkan parameter yang diterima. Program melakukan linear search untuk mencari objek contact yang akan dikembalikan. Jika program tidak menemukan objek contact dengan nama yang sama, maka akan mengembalikan null.
53-72	Method displayAllContact untuk mencetak semua objek contact yang ada dalam ArrayList contact. Pertama, program memeriksa apakah ArrayList contact tidak kosong, jika tidak kosong, program mencetak nama contact dan nomor urutnya. Jika kosong program mencetak "You have no contact."

Main.java

No.	Deskripsi
1	Baris ini menyatakan tempat kelas Main berada.
3	Mengimpor kelas Scanner dari paket java.util. Kelas Scanner digunakan untuk mendapatkan input dari pengguna.
4	Mengimpor semua kelas yang ada di dalam folder Contacts
6	Membuat class Main.
7	Membuat objek scanner untuk secara statis agar dapat diakses diseluruh file Main.
9	Method main dari kelas Main
10	Membuat objek contactList dari class ContactList
12	Blok while yang akan berhenti ketika terdapat keyword break
13	Mencetak pilihan menu program dengan method displayMenu
14-15	Mendapatkan input pilihan menu yang dipilih pengguna. Setelah memanggil scanner.nextInt(), kursor input berada tepat setelah angka terakhir yang dimasukkan oleh pengguna, tetapi sebelum karakter baris baru (\n). Jadi, scanner.nextLine() mengambil string kosong tersebut agar pengguna dapat memasukkan teks selanjutnya saat dibutuhkan.

16-17	Blok switch yang akan berjalan jika pilihan pengguna bukan 0 yaitu pilihan untuk keluar program
18-22	Jika pengguna memilih 1, mencetak semua nama contact yang ada pada ArrayList contact dengan nomor urut lalu menunggu konfirmasi pengguna untuk kembali ke menu.
23-33	Jika pengguna memilih 2, Menampilkan pilihan jenis tipe kontak yang ingin digunakan, lalu menambahkan objek contact ke dalam ArrayList contact dengan informasi dari input pengguna
34-46	Jika pengguna memilih 3 untuk mengubah informasi contact, program meminta input nama dari contact yang ingin diubah, lalu program mendapatkan objek contact dengan method <code>getContactByName</code> . Jika objek contact ada, maka program meminta informasi baru dari contact, Lalu mengubah informasi contact dengan informasi baru.
47-52	Jika pengguna memilih 4 untuk menghapus contact, program meminta input nama dari contact yang ingin diubah, lalu program menghapus objek contact tersebut dari ArrayList contact
53-60	Jika pengguna memilih 5 untuk mencetak detail informasi objek, program meminta input nama dari contact, lalu mencetak nama, nomor telepon dan email dari contact.
61-62	Blok default yang akan berjalan jika input bukan 1-5
64-65	Memberhentikan loop jika pilihan pengguna adalah 0
68	Menutup scanner yang sudah tidak digunakan untuk mencegah <i>resource leak</i>
71-80	Method <code>displayMenu</code> untuk mencetak pilihan menu program
82-106	Method <code>readContact</code> yang mengembalikan objek <code>Contact</code> , <code>PersonalContact</code> atau <code>BusinessContact</code> sesuai dengan informasi dari input pengguna.