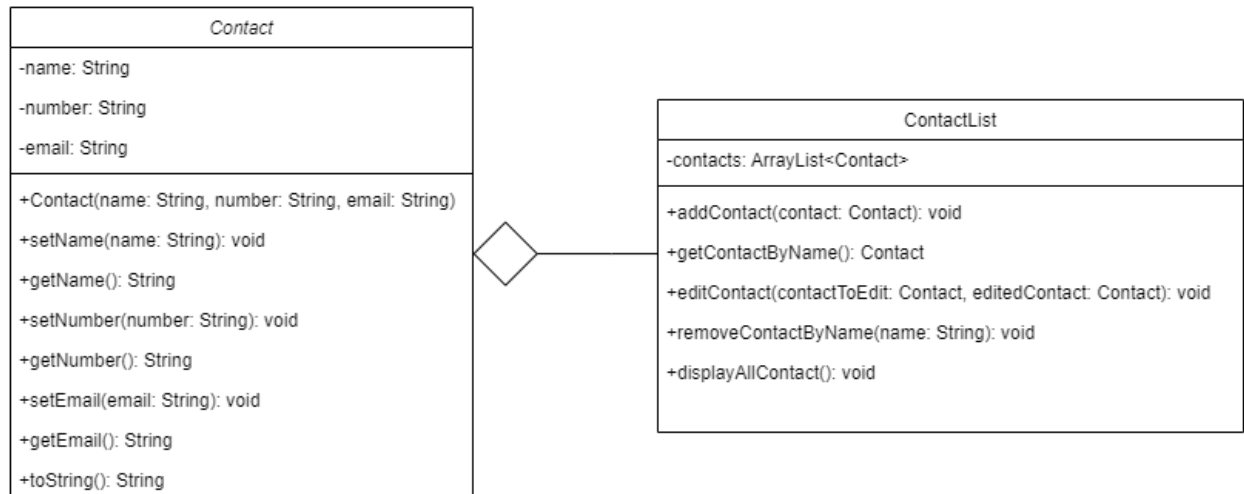


Laporan Praktikum Minggu 3
 Nama: Muhammad Iqbal Alexandre Saputra
 NIM: 11231056
 Kelas: PBO B

Class Diagram Manajemen Kontak



Source Code

No.	org/tugas/tugas3/Contact.java
1.	<code>package org.tugas.tugas3;</code>
2.	
3.	<code>public class Contact {</code>
4.	<code> private String name, number, email;</code>
5.	
6.	<code> public Contact(String name, String number, String email) {</code>
7.	<code> this.name = name;</code>
8.	<code> this.number = number;</code>
9.	<code> this.email = email;</code>
10.	<code> }</code>
11.	
12.	<code> public String getName() {</code>
13.	<code> return name;</code>
14.	<code> }</code>
15.	
16.	<code> public void setName(String name) {</code>
17.	<code> this.name = name;</code>
18.	<code> }</code>
19.	
20.	<code> public String getNumber() {</code>
21.	<code> return number;</code>
22.	<code> }</code>
23.	
24.	<code> public void setNumber(String number) {</code>
25.	<code> this.number = number;</code>
26.	<code> }</code>
27.	
28.	<code> public String getEmail() {</code>

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 +
38.	"\nEmail: " + this.email;
39.	}

No	org/tugas/tugas3/ContactList.java
1.	package org.tugas.tugas3;
2.	
3.	import java.util.ArrayList;
4.	
5.	public class ContactList {
6.	private ArrayList<Contact> contacts = new ArrayList<>();
7.	
8.	public void addContact(Contact contact) {
9.	this.contacts.add(contact);
10.	}
11.	
12.	public void editContact(Contact contactToEdit, Contact editedContact) {
13.	for (Contact contact : this.contacts) {
14.	if (contactToEdit.equals(contact)) {
15.	contact.setName(editedContact.getName());
16.	contact.setNumber(editedContact.getNumber());
17.	contact.setEmail(editedContact.getEmail());
18.	}
19.	}
20.	}
21.	
22.	public void removeContactByName(String name) {
23.	for (Contact contact : this.contacts) {
24.	if (name.equals(contact.getName())) {
25.	this.contacts.remove(contact);
26.	break;
27.	}
28.	}
29.	}
30.	
31.	public Contact getContactByName(String name) {
32.	for (Contact contact : this.contacts) {
33.	if (contact.getName().equals(name)) {
34.	return contact;
35.	}
36.	}

37.	return null;
38.	}
39.	
40.	public void displayAllContact() {
41.	if (this.contacts.size() > 0) {
42.	int counter = 1;
43.	
44.	for (Contact contact : this.contacts) {
45.	System.out.println(counter + ". " + contact.getName());
46.	counter++;
47.	}
48.	} else {
49.	System.out.println("You have no contact.");
50.	}
51.	
52.	}
53.	}

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

```

38.         String name = scanner.nextLine();
39.
40.         contactList.removeContactByName(name);
41.         break;
42.     case 5:
43.         System.out.print("Enter name: ");
44.         name = scanner.nextLine();
45.
46.     System.out.println(contactList.getContactByName(name).toString());
47.         System.out.print("Press enter to continue");
48.         scanner.nextLine();
49.         break;
50.     default:
51.         break;
52.     }
53.     } else {
54.         break;
55.     }
56. }
57. scanner.close();
58. }
59.
60. private static void displayMenu() {
61.     System.out.println("::: Contact Menu :::");
62.     System.out.println("1. Show all contacts");
63.     System.out.println("2. Add a contact");
64.     System.out.println("3. Edit a contact");
65.     System.out.println("4. Delete a contact");
66.     System.out.println("5. Show contact detail");
67.     System.out.println("0. Exit");
68.     System.out.print("Choose one: ");
69. }
70.
71. private static Contact readContact() {
72.     System.out.print("Enter name: ");
73.     String name = scanner.nextLine();
74.
75.     System.out.print("Enter phone number: ");
76.     String phoneNum = scanner.nextLine();
77.
78.     System.out.print("Enter email: ");
79.     String email = scanner.nextLine();
80.
81.     return new Contact(name, phoneNum, email);
82. }
83.
84. }

```

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: 2  
Enter name: Muhammad Iqbal  
Enter phone number: 0887435624959  
Enter email: 11231056@student.itk.ac.id
```

```
::: 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. Muhammad Iqbal  
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 Iqbal  
Enter new name: Muhammad Iqbal A. S.  
Enter new phone number: 0887435624959  
Enter new email: 11231056@student.itk.ac.id
```

```
::: 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: Muhammad Iqbal A. S.  
Name: Muhammad Iqbal A. S.  
Phone Number: 0887435624959  
Email: 11231056@student.itk.ac.id  
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: Muhammad Iqbal A. S.  
::: 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
```

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.

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.
5	Membuat class ContactList.
6	Membuat objek ArrayList untuk menyimpan objek-objek contact.
8-10	Method addContact yang menerima parameter objek Contact untuk menambahkan objek contact ke ArrayList contact.
12-20	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, lalu mengubah atribut objek contact tersebut dengan objek

	contact yang telah diedit menggunakan metode setter.
22-29	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.
31-38	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.
40-50	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.
5	Membuat class Main.
6	Membuat objek scanner untuk secara statis agar dapat diakses diseluruh file Main.
8	Method main dari kelas Main
9	Membuat objek contactList dari class ContactList
11	Blok while yang akan berhenti ketika terdapat keyword break
12	Mencetak pilihan menu program
13-14	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.
15-16	Blok switch yang akan berjalan jika pilihan pengguna bukan 0 yaitu pilihan untuk keluar program
17-21	Jika pengguna memilih 1, mencetak semua nama contact yang ada pada ArrayList contact dengan nomor urut lalu menunggu konfirmasi pengguna untuk kembali ke menu.
22-24	Jika pengguna memilih 2, Menambahkan contact ke dalam ArrayList contact dengan informasi dari input pengguna

25-35	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.
36-41	Jika pengguna memilih 4 untuk menghapus contact, program meminta input nama dari contact yang ingin diubah, lalu program menghapus objek contact tersebut dari <code>ArrayList</code> contact
43-48	Jika pengguna memilih 5 untuk mencetak detail informasi objek, program meminta input nama dari contact, lalu mencetak nama, nomor telepon dan email dari contact.
50-51	Blok default yang akan berjalan jika input bukan 1-5
53-54	Memberhentikan loop jika pilihan pengguna adalah 0
57	Menutup scanner yang sudah tidak digunakan untuk mencegah <i>resource leak</i>
60-69	Method <code>displayMenu</code> untuk mencetak pilihan menu program
71-82	Method <code>readContact</code> yang mengembalikan objek contact dengan informasi dari input pengguna.