1. a)

Truk	Kapal	Pesawat
+ pilihDriver()	+ pilihDriver()	+ pilihDriver()
+ nyalakanMesin()	+ nyalakanMesin()	+ nyalakanMesin()
		+ terbang()

SepedaMotor	Helikopter
+ pilihDriver()	+ pilihDriver()
+ nyalakanMesin()	+ nyalakanMesin()
	+ terbang()

Burung	Ular	Ikan
+ makan()	+ makan()	+ makan()
+ bertelur()	+ bertelur()	+ bertelur()
+ terbang()		

b)
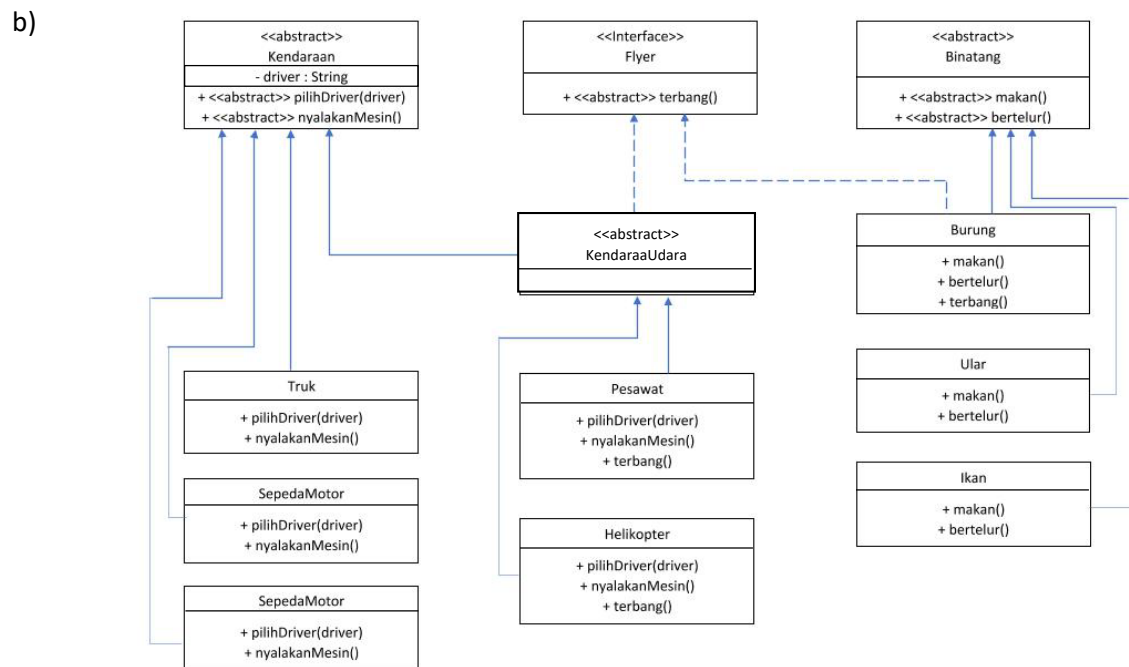
2. a)
```
List <Item> rakBuku = new ArrayList<>();
rakBuku.add(buku);
rakBuku.add(jurnal);
rakBuku.add(icicos);
```

b)
```
for (Item item : rakBuku){
    if (item instanceof Jurnal){
            item.hitungBiayaSewa();
    }
}
```

3.
```
try {
    // Statement
} catch (NumberFormatException  e) {

} catch (ArithmaticException  e) {

} catch (FileNotFoundException  e) {

} catch (IOException  e) {

} catch (Exception  e) {

}
```

4. a)
Project
    Sources Packages
        project.dao.client
            Main.java
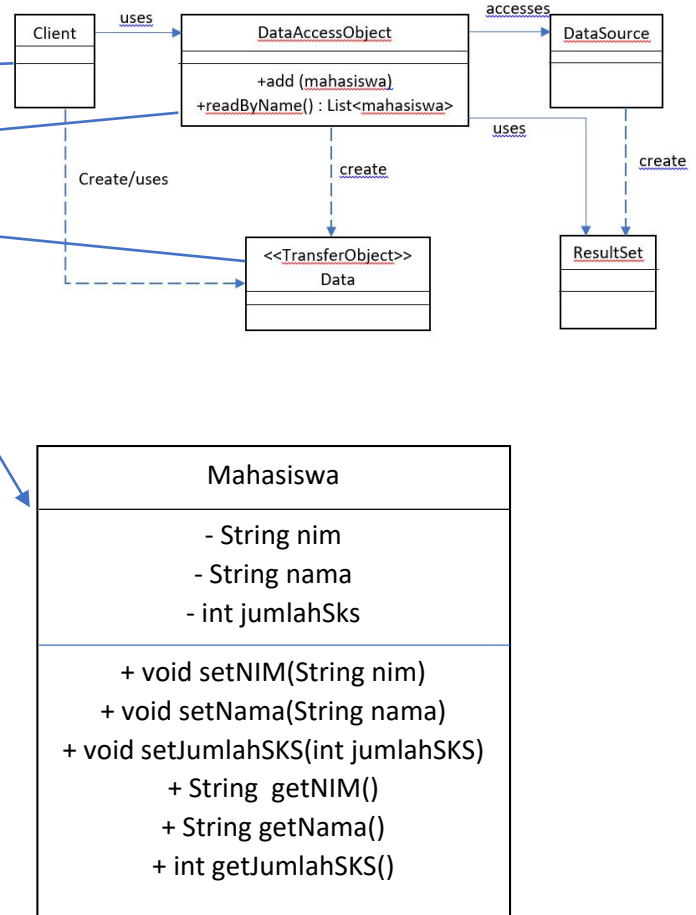   project.dao.dao
            MysqlMahasiswaDAO.java
        project.dao.transferObject
            Mahasiswa.java
        project.utilites
            MysqlUtility.java

    Libraries
        MySQL JDBC Driver
        JDK 1.8

Diagram:

Client --uses--> DataAccessObject --accesses--> DataSource

DataAccessObject:
+add (mahasiswa)
+readByName() : List<mahasiswa>

DataAccessObject --create--> <<TransferObject>> Data

DataSource --uses--> ResultSet (create)
DataSource --create--> ResultSet

Client --Create/uses--> <<TransferObject>> Data

Mahasiswa

- String nim
- String nama
- int jumlahSks

+ void setNIM(String nim)
+ void setNama(String nama)
+ void setJumlahSKS(int jumlahSKS)
+ String getNIM()
+ String getNama()
+ int getJumlahSKS()

```java
b) public void insert(mahasiswa mhs){
        Statement statement = null;

        try {
                statement = koneksi.createStatement();
                String sql = "INSERT into mahasiswa(nim, nama, jumlahSKS) VALUES(' " + mhs.getNIM() +
        "','" + mhs.getNama() + "'," + man.getJumlahSKS() +"')";
                if (statement.executeUpdate(sql) > 0) {
                    System.out.println("Berhasil insert");
                } else {
                    System.out.println("Gagal insert");
                }
            } catch (SQLException ex) {
                System.out.println("SQLException: " + ex.getMessage());
            } finally {
                if (statement != null) {
                    try {
                        statement.close();
                    } catch (SQLException ex) {
                        System.out.println("Gagal menutup statement");
                    }
                }
            }
    }
```

```java
public List<mahasiswa> selectByNama() {
    Statement statement = null;
    ResultSet result = null;
    List<mahasiswa> list = new ArrayList<>();
    try {
        statement = koneksi.createStatement();
        String sql = "SELECT * FROM mahasiswa ORDER nama DESC";
        result = statement.executeQuery(sql);

        while (result.next()) {
            mahasiswa mhs = new mahasiswa("", "", 0);
            mhs.setNIM(result.getString("nim"));
            mhs.setNama(result.getString("nama"));
            mhs.setJumlahSKS(result.getInt("jumlahSKS"));
            list.add(mhs);
        }
        return list;

    } catch (SQLException ex) {
        System.out.println("SQLException: " + ex.getMessage());
        System.out.println("Objek Statement berhasil dibuat");
        System.out.println("Objek Statement gagal dibuat");
        return null;
    } finally {
        if (statement != null) {
            try {
                statement.close();
            } catch (SQLException ex) {
                System.out.println("Gagal menutup statement");
            }
        }
        if (result != null) {
            try {
                result.close();
            } catch (SQLException ex) {
                System.out.println("ResultSet gagal ditutup");
            }
        }
    }
}
```

Nb : jawaban belum tentu 100% benar, maaf jika ada kesalahan