



BAB II

TINJAUAN PUSTAKA

2.1 Teori Judul

2.1.1 Pengertian Aplikasi

Menurut Jogiyanto (2012), Aplikasi adalah sekelompok atribut yang terdiri dari beberapa form, report yang disusun sedemikian rupa sehingga dapat mengakses data. Aplikasi merupakan program yang berisikan perintah-perintah untuk melakukan pengolahan data. Jogiyanto menambahkan aplikasi secara umum adalah suatu proses dari cara manual yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal.

Menurut Elisa (2016), Aplikasi sering juga disebut sebagai perangkat lunak, merupakan program komputer yang isi instruksinya dapat diubah dengan mudah.

2.1.2 Pengertian Pemesanan

Menurut Utara (2011), pemesanan adalah suatu aktivitas yang dijalankan oleh konsumen sebelum membeli. Untuk mewujudkan kepuasan konsumen maka perusahaan harus mempunyai sebuah sistem pemesanan yang baik. Tujuan pemesanan yaitu :

1. Memaximumkan pelayanan bagi konsumen.
2. Meminimumkan investasi pada persediaan.
3. Perencanaan kapasistas
4. Persediaan dan kapasitas. dan lain-lain.

Menurut Septian (2017), Pemesanan dalam arti umum adalah perjanjian antara 2 pihak atau lebih, perjanjian tersebut dapat berupa produk atau jasa.

2.1.3 Pengertian Bahan Bakar Minyak (BBM)

Menurut sawitri (2015), Bahan bakar minyak (BBM) sangat penting dalam kehidupan masyarakat. BBM merupakan kebutuhan pokok bagi masyarakat desa maupun kota baik sebagai rumah tangga maupun sebagai pengusaha, demikian



juga BBM sangat penting bagi sector industri maupun transportasi. Oleh karena begitu pentingnya BBM dalam kehidupan masyarakat, maka BBM termasuk salah satu kebutuhan pokok masyarakat.

Sesuai Peraturan Presiden No. 191/2014 terdapat tiga jenis bahan bakar minyak yang ditetapkan oleh Pemerintah.

Adapun ketiga jenis BBM tersebut antara lain:

1. Jenis BBM Tertentu (JBT). BBM jenis ini disubsidi oleh pemerintah dan didistribusikan ke seluruh wilayah Indonesia. Adapun produknya adalah Minyak Tanah dan Minyak Solar.
2. Jenis BBM Khusus Penugasan (JBKP). BBM jenis ini Non Subsididan hanya didistribusikan di wilayah penugasan selain Pulau Jawa, Madura, dan Bali (Jamali) sesuai Perpres No. 191/2014. Adapun produk BBM ini adalah Premium.
3. Jenis BBM Umum (JBU). BBM Non Subsidi ini didistribusikan ke seluruh wilayah Indonesia (kecuali premium Jamali). Adapun produknya adalah Premium Jamali, Perta Series (Pertalite, Pertamina, Pertamina Turbo) dan Dex Series (Dexlite, Pertamina Dex).

Pada pelaksanaan distribusi, harga produk non komersil Minyak Tanah, Solar, Premium non Jamali ditentukan oleh Pemerintah.

Sementara produk komersil yakni Premium Jamali, Perta Series, Dex Series harganya ditetapkan oleh Badan Usaha.

2.1.4 Pengertian Aplikasi Pemesanan *Online* pada PT. Reza Sentosa Perkasa

Aplikasi pemesanan *online* pada PT. Reza Sentosa Perkasa merupakan aplikasi yang dibuat untuk mempermudah pelanggan dalam melakukan pesanan terhadap produk yang ada di PT. Reza Sentosa Perkasa secara *online*.



2.2 Teori Umum

2.2.1 PHP

Menurut sidik (2017), PHP merupakan secara umum dikenal sebagai bahasa pemrograman script script yang membuat dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor HTML. Dikenal juga sebagai bahasa pemrograman server side.

Dengan menggunakan PHP maka maintenance suatu situs web menjadi lebih mudah. Proses update data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan script PHP.

2.2.2 MySql

Menurut sidik (2017), MySql merupakan software database yang termasuk paling populer di lingkungan Linux, kepopuleran ini karena ditunjang karena performansi query dari databasesnya yang saat itu bisa dikatakan paling cepat, dan jarang bermasalah. MySql telah tersedia juga dilingkungan Windows.

Berangkat dari software yang shareware MySql populer, kini mulai versi 3.23 MySql menjadi software open source yang berarti free. MySql dapat digunakan untuk kepentingan komersial atau pun personal (non profit).

2.2.3 Android

Android adalah sistem operasi berbasis *Linux* yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh *Android Inc* dengan dukungan finansial dari Google, yang kemudian Google membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008.

Menurut Arifianto.T (2011:1), Android adalah sebuah nama untuk system operasi pada suatu gadget seperti computer tablet, smartphone, dan telephone



cellular. Sistem operasi yang digunakan berbasis Linux yang dikembangkan oleh Google Inc. Simbol/ logi android berbentuk robot dengan dua antenna dikepalanya, ini melambangkan bahwa android merupakan simbolisasi dari system operasi kelas atas untuk gadget dan smartphone.

Dari awal peluncurannya pada tahun 2007, Android sudah beberapa kali melakukan pembaruan versinya. Versi pada Android memiliki nama yang unik (kebanyakan nama makanan manis), antara lain :

- a. Android versi 1.1
- b. Android versi 1.5 (Cupcake)
- c. Android versi 1.6 (Donut)
- d. Android versi 2.0/ 2.1 (Éclair)
- e. Android versi 2.2 (Froyo: Frozen Yoghurt)
- f. Android versi 2.3 (Gingerbread)
- g. Android versi 3.0/ 3.1 (Honeycomb)
- h. Android versi 4.0 (Ice Cream Sandwich)
- i. Android versi 4.1/ 4.2 (Jelly Bean)
- j. Android versi 4.4 (Kitkat)

2.2.4 Eclipse

Menurut Nasruddin Safaat h (2012:16) Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platformindependent*).

Berikut ini adalah sifat dari Ec lipse:

- a) Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linu x, Solaris, AIX, HP-UX dan Mac OS X.
- b) Multi-language: Eclipse dike mbangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol,



Python, Perl, PHP, dan lain sebagainya.

- c) Multi-role: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plugin. Eclipse dibuat dari kerja sama antara perusahaan-perusahaan anggota '*Eclipse Foundation*' (berserta individu-individu lain). Banyak nama besar yang ikut dalam '*Eclipse Foundation*', termasuk IBM, BEA, Intel, Nokia, Borland. Eclipse bersaing langsung dengan Netbeans IDE. Plugin tambahan pada Eclipse jauh lebih banyak dan bervariasi dibandingkan IDE lainnya.

2.2.5 Google Maps

Google Maps adalah sebuah jasa peta globe virtual gratis dan online yang disediakan oleh Google dan dapat ditemukan di <https://maps.google.com>. (Wikipedia.org). Google maps menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia, dan baru-baru ini bulan, dan juga menawarkan perencanaan rute dan pencari letak bisnis di US, Kanada, Jepang, Hong Kong, Cina, UK, Irlandia (hanya pusat kota), dan beberapa bagian Eropa. Google Maps masih berada dalam tahap beta.

2.3 Teori Khusus

2.3.1 Object Oriented Program (OOP)

Menurut Abdullah (2017), OOP (*Object Oriented Program*) merupakan teknik pemrograman dengan menggunakan konsep objek. Tujuan dari OOP adalah untuk memudahkan programmer dalam pembuatan program dengan menggunakan konsep objek yang ada dalam kehidupan sehari-hari. Jadi setiap bagian



permasalahan adalah objek, dan objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil.

Sebuah objek pada OOP memiliki data atau disebut property yang menjelaskan tentang sifat-sifat objek tersebut. Seperti sebuah handphone dapat memiliki data warna, merk, ukuran layar, dan sebagainya. Begitu juga dengan objek-objek yang ada didalamnya seperti layar memiliki data berupa lebar, tinggi dan sebagainya.

Selain memiliki property, sebuah objek dalam OOP memiliki method berupa fungsi yang dapat dipanggil untuk melakukan tindakan atau merubah nilai dari property yang ada di dalamnya. Seperti handphone dapat melakukan tindakan merekam, restart, memanggil, mengirim pesan dan sebagainya. Handphone juga dapat diganti *casing* untuk mengubah warnanya (mengubah nilai property).

2.3.2 *Unified Modelling Language (UML)*

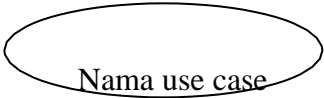
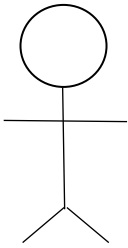

Menurut Sukamto (2018:13), *Unified Modeling Language (UML)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

2.3.3 *Usecase Diagram*

Sukamto (2018:155), *use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada

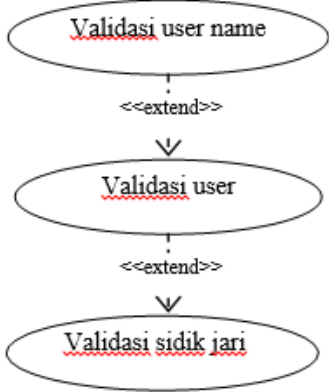

pada diagram *use case* :

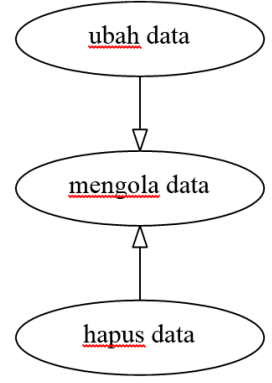
Tabel 2.1 Simbol-simbol diagram *use case*

No.	Simbol	Deskripsi
1.	<i>Use case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .
2.	Aktor/ <i>actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	Assosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.	Exstensi/ <i>extend</i> <<exten d>>	Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi

	objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, missal
--	--

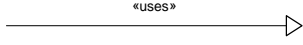
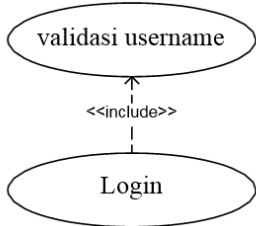
Lanjutan Tabel 2.1 Simbol-simbol diagram *use case*

No.	Simbol	Deskripsi
		 <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
5.	<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua <i>buah use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :</p>

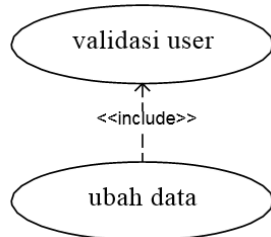
		 <p>arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
--	--	--

Lanjutan Tabel 2.1 Simbol-simbol diagram *use case*

No.	Simbol	Deskripsi
6.	Menggunakan / <i>include</i> / <i>uses</i> <<include>>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> : <ul style="list-style-type: none"> - <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, missal pada kasus berikut :

		 <p>- <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan di jalankan, misal pada kasus berikut :</p>
--	---	--

Lanjutan Tabel 2.1 Simbol-simbol diagram *use case*



No.	Simbol	Deskripsi
		 <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber : Sukamto (2018:156-158)



2.3.4 Activity Diagram


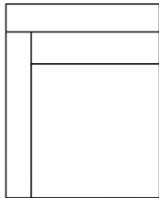
Sukamto (2018:161), diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Tabel 2.2 Simbol-simbol *activity diagram*

No.	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.

Lanjutan Tabel 2.2 Simbol-simbol *activity diagram*

No.	Simbol	Deskripsi
3.	Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

5.	Status akhir 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

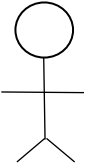
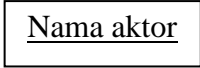

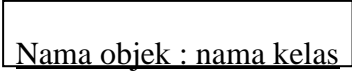

Sumber : Sukamto (2018:162-163)

2.3.5 *Sequence Diagram*

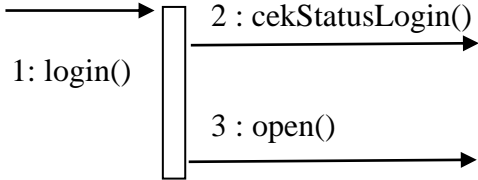
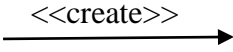
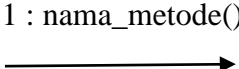
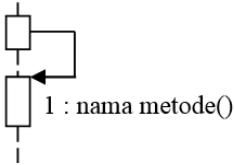

Sukamto (2018:165), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan

diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya jalannya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Berikut adalah simbol-simbol yang ada pada diagram sekuen :


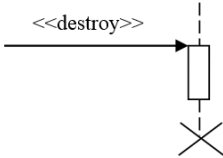
Tabel 2.4. Simbol-simbol *sequence diagram*

No.	Simbol	Deskripsi
1.	Aktor  Atau  Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.
2.	Garis hidup/ <i>lifeline</i> 	Menyatakan kehidupan suatu objek
3.	Objek 	Menyatakan objek yang berinteraksi pesan
4.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya

Lanjutan Tabel 2.4. Simbol-simbol *sequence diagram*

No.	Simbol	Deskripsi
		<p>Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif</p>
5.	<p>Pesan tipe <i>create</i></p> <p style="text-align: center;">  </p>	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
6.	<p>Pesan tipe <i>call</i></p> <p style="text-align: center;">  </p>	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p> <p style="text-align: center;">  </p> <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
7.	<p>Pesan tipe <i>send</i></p> <p style="text-align: center;">  </p>	<p>Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>

Lanjutan Tabel 2.4. Simbol-simbol *sequence diagram*

No.	Simbol	Deskripsi
8.	<p>Pesan tipe <i>return</i></p> <p>1 : keluaran</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
9.	<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>Destroy</i></p>

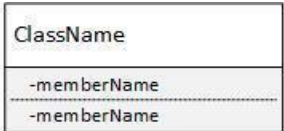
Sumber: : Sukamto (2018:165-167)

2.3.6 Class Diagram




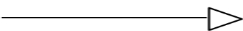
Sukamto (2018:141), diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan atribut dan *method* :

1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

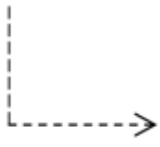
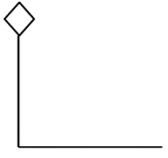
Tabel 2.3 Simbol-simbol *class diagram*

No.	Simbol	Deskripsi
1.	<p>Kelas</p> 	<p>Kelas pada struktur sistem</p>



2.	Antarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.	Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum - khusus)

Lanjutan Tabel 2.3 Simbol-simbol *class diagram*

No.	Simbol	Deskripsi
6.	Kebergantungan/ <i>dependensi</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
7.	Agregasi/ <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

Sumber : Sukamto (2018:146-147)