CHAPTER IV

DATA COLLECTION & PROCESSING

Research Thinking Flow below will be shown in Figure 4.1 the main step of the research to understand this research easier and correctly.

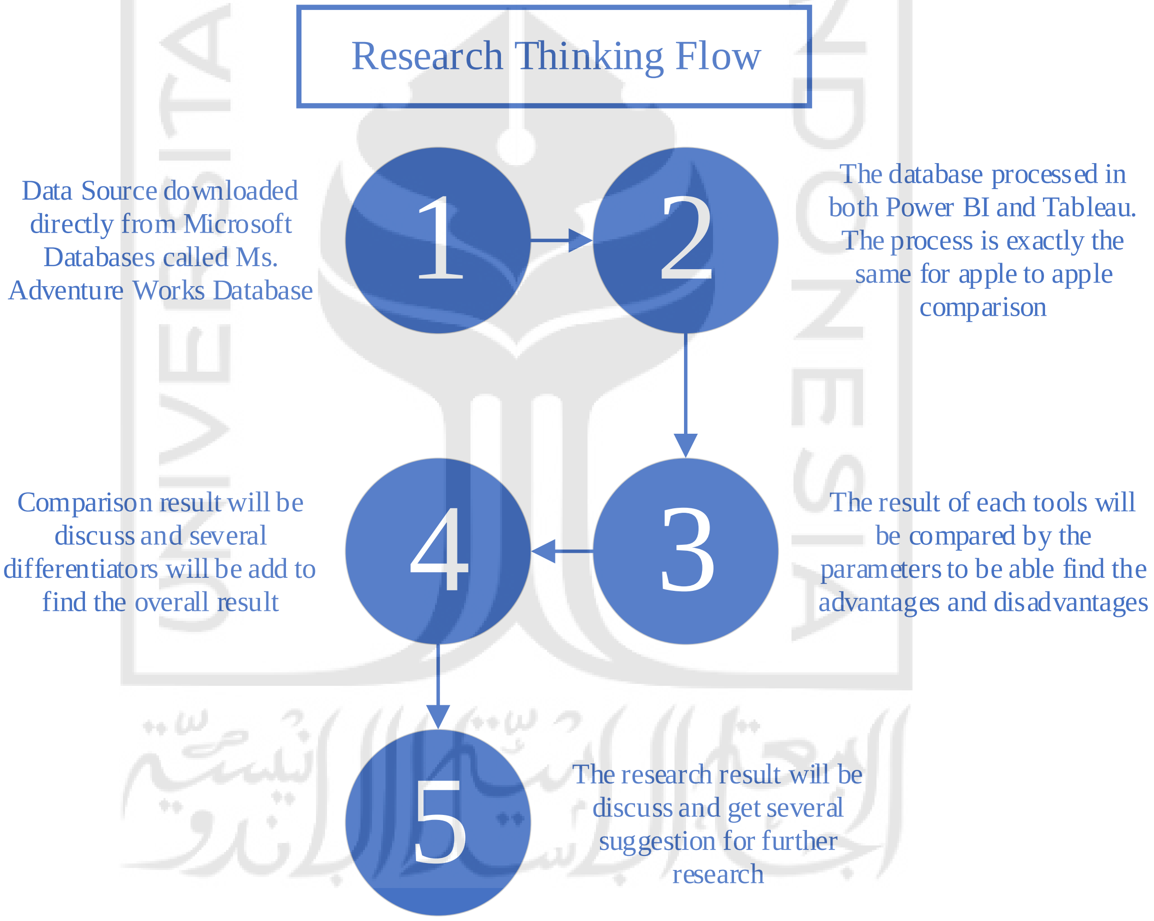

Research Thinking Flow
1
Data Source downloaded directly from Microsoft Databases called Ms. Adventure Works Database
2
The database processed in both Power BI and Tableau. The process is exactly the same for apple to apple comparison
3
The result of each tools will be compared by the parameters to be able find the advantages and disadvantages
4
Comparison result will be discuss and several differentiators will be add to find the overall result
5
The research result will be discuss and get several suggestion for further research

Figure 4.1 Research Thinking Flow

4.1 Data Collection

This chapter explains the process of data collection. The researcher got the secondary data from the Microsoft website called Microsoft Adventure Works Database. Microsoft Adventure Works is a fictional bicycle wholesaler company. The company has 97 different brands of bikes that grouped into three categories: mountain bikes, road bikes, and touring bikes. Moreover, Adventure Works also manufactures some of its own components. Several components, accessories, and clothing are purchased from outside from vendors.

4.1.1 Company Database Profile

Adventure Works is not only selling bicycles, but it also provides accessories, clothing, and components. The accessories available such as bottles, bike racks, brakes, etc. The clothing such as caps, gloves, jersey, etc. For the components, Adventure Works sells brakes, chains, derailleurs, etc. Many of those things are made by vendors, so Adventure Works stands as a reseller.

Adventure Works serve the customer globally such as Australia, Canada, France, Germany, the United Kingdom, and the United States. There are 2 business models in Adventure Works which are retail stores that sell bikes, and internet sales that serve individual customers. Usually, Adventure Works sells in bulk to retail stores, which acts as resellers for its products.

To run the business activities, Adventure Works has a total of 290 employees that included in some functions such as sales, production, purchasing, engineering, finance, information services, marketing, shipping and receiving, and R&D. The customers of Adventure Works include over 700 stores and over 19000 individuals worldwide and its vendors are quantified around 100 vendors companies that supply raw materials, accessories, clothing, and components.

Even though Adventure Works is fictional, it is designed as a realistic case as the same as a real company in the industry. Adventure Works provide database and data warehouse that covers the business process from sales, material management, production, finance, and human capital management. Therefore, the researcher uses this fictional company as the case study to simulate a comparison study of BI system in Power BI and Tableau Software.

4.1.2 Adventure Works 2017 List Details

This database contains 68 tables from company transactions. There are several tables that grouped into a different area such as sales, human resources, person, purchasing, and production. Table 4.1 shows the list of tables stored in Adventure Works 2017.

Table 4.1 Adventure Works 2017 List

No	Table Name	Descriptions
1	Address	Street address information for customers, employees, and vendors
2	AddressType	Types of addresses stored in the Address table.
3	BillOfMaterials	Items required to make bicycles and bicycle subassemblies.
4	BusinessEntity	Source of the ID that connects vendors, customers, and employees with address and contact information
5	BusinessEntityAddress	Cross-reference table mapping customers, vendors, and employees to their addresses.
6	BusinessEntityContact	Cross-reference table mapping stores, vendors, and employees to people
7	ContactType	Lookup table containing the types of business entity contacts.
8	CountryRegion	Lookup table containing the ISO standard codes for countries and regions.
9	CountryRegionCurrency	Cross-reference table mapping ISO currency codes to a country or region.
10	CreditCard	Customer credit card information
11	Culture	Lookup table containing the languages in which some Adventure Works data is stored
12	Currency	Lookup table containing standard ISO currencies.
13	CurrencyRate	Currency exchange rates.
14	Customer	Current customer information. Also, see the Person and Store tables
15	Department	Lookup table containing the departments within the Adventure Works Cycles company.
16	Document	Product maintenance documents.
17	EmailAddress	Where to send a person email
18	Employee	Employee information such as salary, department, and title.
19	EmployeeDepartmentHistory	Employee department transfers.
20	EmployeePayHistory	Employee pay history.
21	Illustration	Bicycle assembly diagrams

No	Table Name	Descriptions
22	JobCandidate	Resumes submitted to Human Resources by job applicants.
23	Location	Product inventory and manufacturing locations
24	Password	One-way hashed authentication information
25	Person	Human beings involved with Adventure Works: employees, customer contacts, and vendor contacts
26	PersonCreditCard	Cross-reference table mapping people to their credit card information in the CreditCard table
27	PersonPhone	Telephone number and type of person.
28	PhoneNumberType	Type of phone number of a person.
29	Product	Products sold or used in the manufacturing of sold products.
30	ProductCategory	High-level product categorization.
31	ProductCostHistory	Changes in the cost of a product over time.
32	ProductDescription	Product descriptions in several languages.
33	ProductDocument	Cross-reference table mapping products to related product documents.
34	ProductInventory	Product inventory information
35	ProductListPriceHistory	Changes in the list price of a product over time.
36	ProductModel	Product model classification.
37	ProductModelIllustration	Cross-reference table mapping product models and illustrations.
38	ProductModel ProductDescriptionCulture	Cross-reference table mapping product descriptions and the language the description is written in.
39	ProductPhoto	Product images.
40	ProductProductPhoto	Cross-reference table mapping products and product photos.
41	ProductReview	Customer reviews of products they have purchased.
42	ProductSubcategory	Product subcategory classification
43	ProductVendor	Cross-reference table mapping vendors with the products they supply.
44	PurchaseOrderDetail	Individual products associated with a specific purchase order. See PurchaseOrderHeader.
45	PurchaseOrderHeader	Individual products associated with a specific purchase order. See PurchaseOrderHeader.
46	SalesOrderDetail	Individual products associated with a specific sales order. See SalesOrderHeader.
47	SalesOrderHeader	General sales order information.
48	SalesOrderHeader SalesReason	Cross-reference table mapping sales orders to sales reason codes
49	SalesPerson	Sales representative current information.
50	SalesPersonQuotaHistory	Sales representative current information.
51	SalesReason	A lookup table of customer purchase reasons.
52	SalesTaxRate	Tax rate lookup table.
53	SalesTerritory	Sales territory lookup table

No	Table Name	Descriptions
54	SalesTerritoryHistory	Sales representative transfers to other sales territories.
55	ScrapReason	Manufacturing failure reasons lookup table.
56	Shift	Work shift lookup table.
57	ShipMethod	Shipping company lookup table.
58	ShoppingCartItem	Contains online customer orders until the order is submitted or canceled.
59	SpecialOffer	Sale discounts lookup table.
60	SpecialOfferProduct	Cross-reference table mapping products to special offer discounts
61	StateProvince	State and province lookup table.
62	Store	Customers (resellers) of Adventure Works products
63	TransactionHistory	Record of each purchase order, sales order, or work order transaction year to date
64	TransactionHistoryArchive	Transactions for previous years.
65	UnitMeasure	Unit of a measure lookup table.
66	Vendor	Companies from whom Adventure Works Cycles purchases parts or other goods.
67	WorkOrder	Manufacturing work orders.
68	WorkOrderRouting	Work order details.

Table 4.1 shows the list of tables stored in the Adventure Works database. These data consist of several classifications such as sales, purchasing, product, human resources, and person. These tables will be used as the main source to create an OLAP cube contains several dimensions with 1 fact table.

4.2 Data Processing

To conduct a comparison study, the researcher should simulate the Self-Service Business Intelligence system to get the apple to apple comparison result. The researcher will implement sales analysis for BI using Microsoft Adventure Works 2017 Database.

4.2.1 Importing Database

To develop a BI system, the first step that should be performed is importing the data or database. In this case, the researcher uses Microsoft SQL Server 2017 to store the Adventure Works 2017 database. The database that already stored in the SQL Server can be used for both Power BI and Tableau Software without interrupting each other. It is because each BI Tools has its own local server to store the data after the connection finish.

4.2.2 Design Data Warehouse

The data warehousing process is a process that is used as basic data for BI solutions. Therefore, it is necessary to design a data warehouse. The design of the data warehouse itself is done by forming a new database as a place for data to be processed to implement BI.

Before the researcher can conduct a comparison study between both software, the researcher must conduct the simulation of real BI Analysis. In this step, the researcher conducts a simple Data Warehouse which is the focus in sales performance in the result as “Product Sales Analysis “. This scope only to decide the last step of BI Analysis and to be set as the goal of conducting the BI.

4.2.2.1 Develop using Power BI

Power BI has its integrated software to develop Data Warehouse inside the Power BI software. This software called Power Query. Power Query editor could conduct the development of Data Warehousing step and do the ETL (Extract, Transform, Load) process continuously.

This editor can also develop a relationship diagram such as common Data Warehouse function in the world market. This tool is very useful because of easier the user to see in the point of eagle eyes view. The design of Power BI Data Warehouse as shown in figure 4.1.

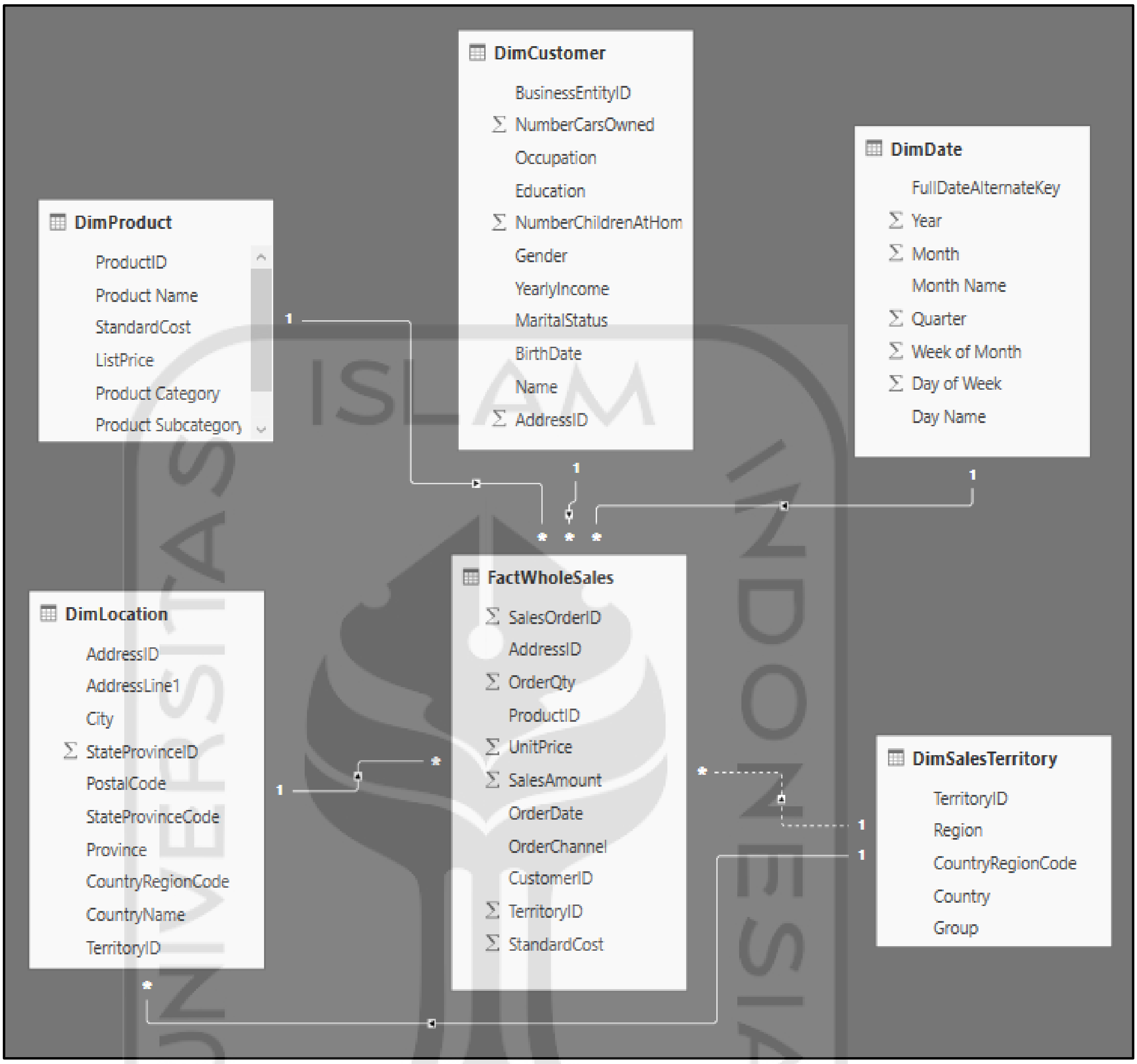


Figure 4.2. Power BI Data Warehouse Diagram

Figure 4.2 shows the Data Warehouse from Power BI design. The design using a star-schema model with 1 fact table. This design will be used as guidance to develop OLAP database. This relationship is created directly in Power Query Editor inside the Power BI.

4.2.2.2 Develop using Tableau Software

Tableau Software or Tableau Desktop is a BI Analysis Tools that doesn’t have their integrated Data Warehouse Tools such as Power Query in Power BI. The common thing that user conduct in Tableau is Business Analytics. Yet, since 2018 Tableau establishes their tools that can be used to prepare our data.

The software called Tableau Prep (Currently change to Tableau Prep Builder). Tableau Prep Builder is a tool in the Tableau product suite designed to prepare the easy and intuitive data. This tool can be used to combine, shape, and clean the data for analysis in Tableau. Tableau Prep can be used to join, union, aggregate, etc.

This tool is quite different than the common Data Warehouse tool. It is because this tool not developed specifically for Data Warehousing. Yet, it can also be applied to analyze the data itself or it could stand alone. This tool creates a flow to transform data. In that flow, the user can easily edit, change, join, union, or etc. their data. It will be shown in Figure 4.2.

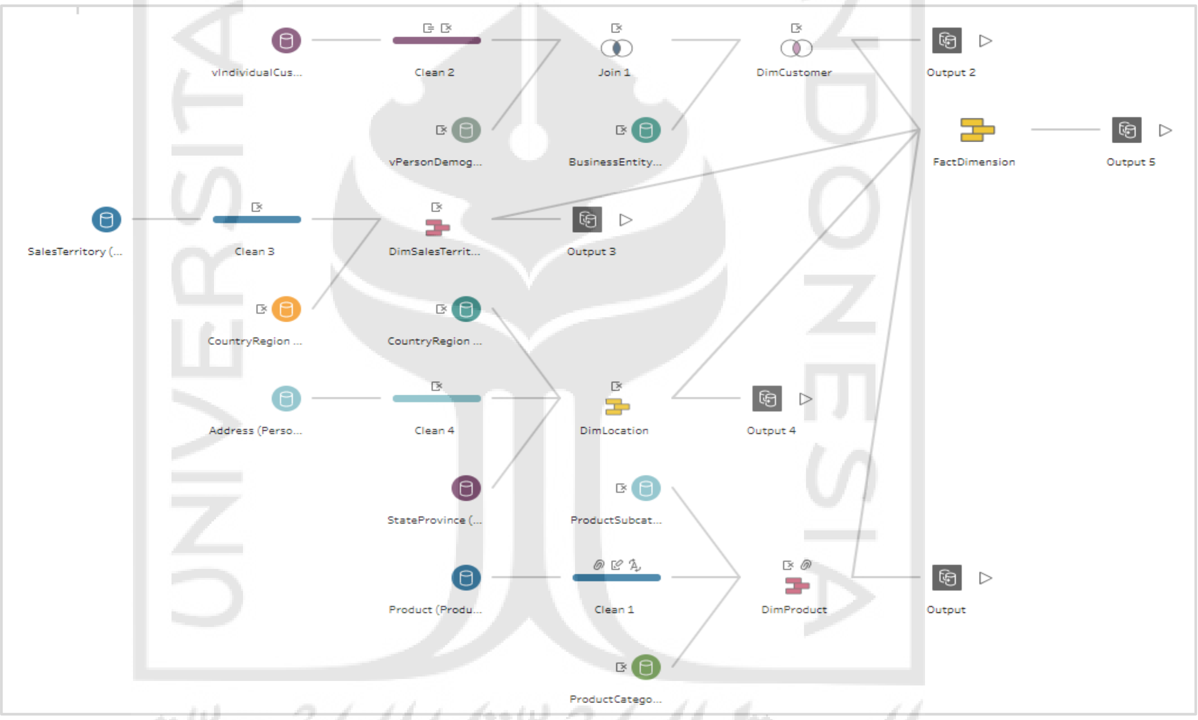


Figure 4.3. Tableau Prep Builder Flow Diagram

Figure 4.3 shows the flow of Tableau Prep Builder. The design is the same, which is using a star-schema model with 1 fact table. This design will be used as guidance to develop OLAP analysis in Tableau Desktop. This flow can directly export and open the Tableau Desktop within a single click.

The similarity between the Power BI/Power Query and Tableau Prep Builder / Tableau Desktop is in both software, the step for design the relationship diagram for developing the Data Warehouse already one with the ETL process. It is different from the other tool such as Microsoft SQL Server Management Studio (SSMS) and Microsoft

Visual Studio or other BI Tools which have specific step and the step is clearly one by one. This both software designed specialized for SSBI user.

4.2.3 ETL Process

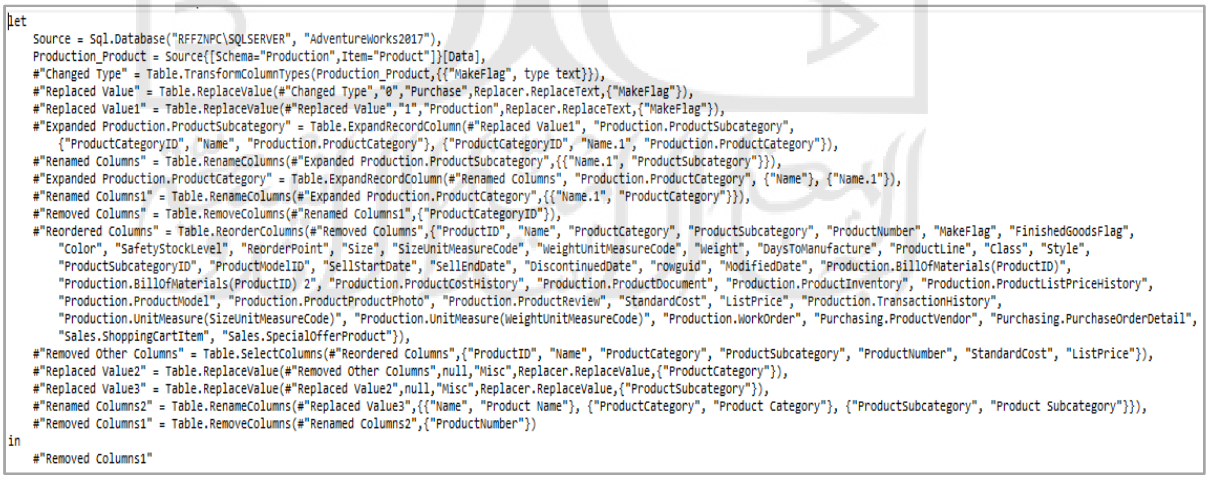
In this section, as the researcher explained the previous section, the ETL process was conducted directly after the data had already been imported from the database. In the Power BI software, the ETL process was conducted directly in the Power Query Editor which already had been integrated with Power BI software. Same as Power BI, Tableau Prep Builder also conducted this step directly after the data have already been imported from the database. The result of ETL in Power BI is a relationship diagram, while the result of Tableau Prep Builder is flow diagram.

The result of the ETL process in Power BI can show the pseudo code and the result table. Yet, the Tableau Prep Builder can only show the result table. Below is the result of each software.

4.2.3.1 Power BI Result

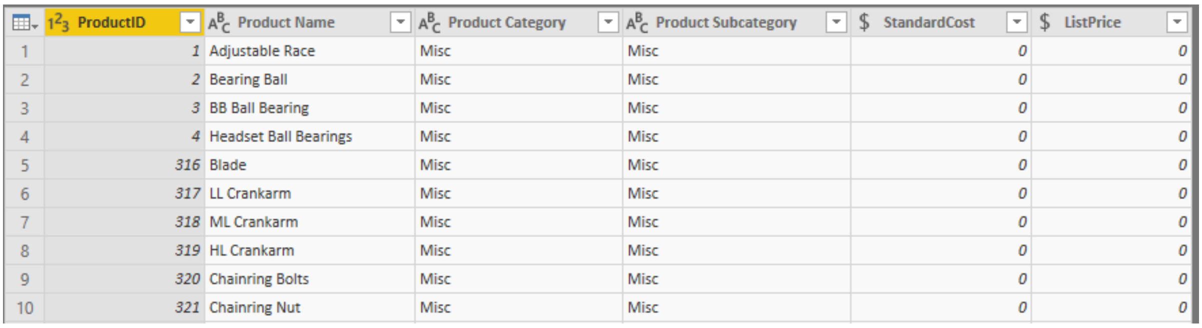
In the Power Query, performing the ETL process based on data warehouse model will be created a relationship diagram. These are the dimensions, the pseudo-code from Power Query and the result table.

1. DimProduct



```
let
    Source = Sql.Database("RFFZNPC\SQLSERVER", "AdventureWorks2017"),
    Production_Product = Source{[Schema="Production",Item="Product"]}[Data],
    #"Changed Type" = Table.TransformColumnTypes(Production_Product,{{"MakeFlag", type text}}),
    #"Replaced Value" = Table.ReplaceValue(#"Changed Type","0","Purchase",Replacer.ReplaceText,{"MakeFlag"}),
    #"Replaced Value1" = Table.ReplaceValue(#"Replaced Value","1","Production",Replacer.ReplaceText,{"MakeFlag"}),
    #"Expanded Production.ProductSubcategory" = Table.ExpandRecordColumn(#"Replaced Value1", "Production.ProductSubcategory",
        {"ProductCategoryID", "Name", "Production.ProductCategory"}, {"ProductCategoryID", "Name.1", "Production.ProductCategory"}),
    #"Renamed Columns" = Table.RenameColumns(#"Expanded Production.ProductSubcategory",{{"Name.1", "ProductSubcategory"}}),
    #"Expanded Production.ProductCategory" = Table.ExpandRecordColumn(#"Renamed Columns", "Production.ProductCategory", {"Name"}, {"Name.1"}),
    #"Renamed Columns1" = Table.RenameColumns(#"Expanded Production.ProductCategory",{{"Name.1", "ProductCategory"}}),
    #"Removed Columns" = Table.RemoveColumns(#"Renamed Columns1",{"ProductCategoryID"}),
    #"Reordered Columns" = Table.ReorderColumns(#"Removed Columns",{"ProductID", "Name", "ProductCategory", "ProductSubcategory", "ProductNumber", "MakeFlag", "FinishedGoodsFlag",
        "Color", "SafetyStockLevel", "ReorderPoint", "Size", "SizeUnitMeasureCode", "WeightUnitMeasureCode", "Weight", "DaysToManufacture", "ProductLine", "Class", "Style",
        "ProductSubcategoryID", "ProductModelID", "SellStartDate", "SellEndDate", "DiscontinuedDate", "rowguid", "ModifiedDate", "Production.BillOfMaterials(ProductID)",
        "Production.BillOfMaterials(ProductID) 2", "Production.ProductCostHistory", "Production.ProductDocument", "Production.ProductInventory", "Production.ProductListPriceHistory",
        "Production.ProductModel", "Production.ProductProductPhoto", "Production.ProductReview", "StandardCost", "ListPrice", "Production.TransactionHistory",
        "Production.UnitMeasure(SizeUnitMeasureCode)", "Production.UnitMeasure(WeightUnitMeasureCode)", "Production.WorkOrder", "Purchasing.ProductVendor", "Purchasing.PurchaseOrderDetail",
        "Sales.ShoppingCartItem", "Sales.SpecialOfferProduct"}),
    #"Removed Other Columns" = Table.SelectColumns(#"Reordered Columns",{"ProductID", "Name", "ProductCategory", "ProductSubcategory", "ProductNumber", "StandardCost", "ListPrice"}),
    #"Replaced Value2" = Table.ReplaceValue(#"Removed Other Columns",null,"Misc",Replacer.ReplaceValue,{"ProductCategory"}),
    #"Replaced Value3" = Table.ReplaceValue(#"Replaced Value2",null,"Misc",Replacer.ReplaceValue,{"ProductSubcategory"}),
    #"Renamed Columns2" = Table.RenameColumns(#"Replaced Value3",{{"Name", "Product Name"}, {"ProductCategory", "Product Category"}, {"ProductSubcategory", "Product Subcategory"}}),
    #"Removed Columns1" = Table.RemoveColumns(#"Renamed Columns2",{"ProductNumber"})
in
    #"Removed Columns1"
```

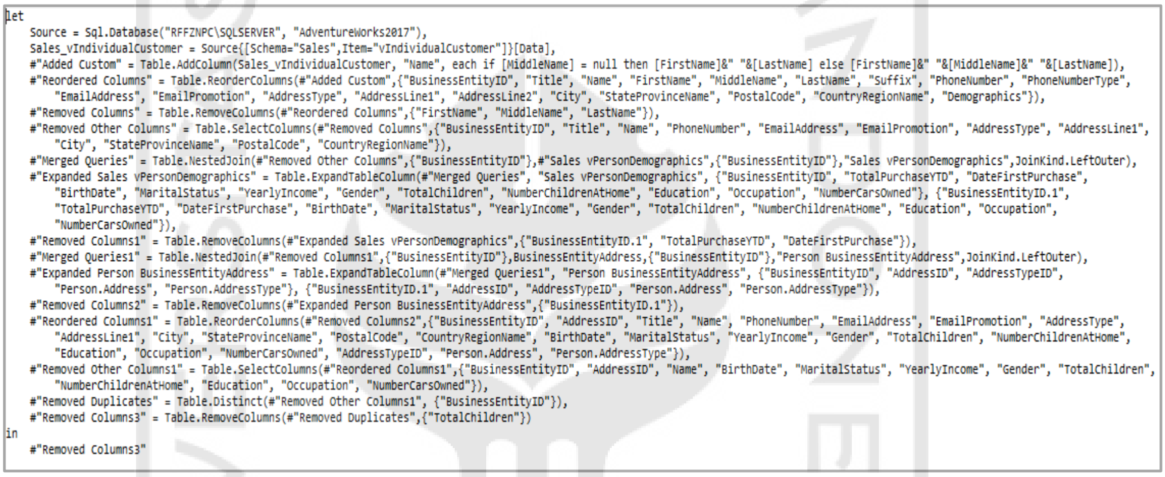
Figure 4.4. Pseudo Code of PowerQuery DimProduct



	ProductID	Product Name	Product Category	Product Subcategory	StandardCost	ListPrice
1	1	Adjustable Race	Misc	Misc	0	0
2	2	Bearing Ball	Misc	Misc	0	0
3	3	BB Ball Bearing	Misc	Misc	0	0
4	4	Headset Ball Bearings	Misc	Misc	0	0
5	316	Blade	Misc	Misc	0	0
6	317	LL Crankarm	Misc	Misc	0	0
7	318	ML Crankarm	Misc	Misc	0	0
8	319	HL Crankarm	Misc	Misc	0	0
9	320	Chainring Bolts	Misc	Misc	0	0
10	321	Chainring Nut	Misc	Misc	0	0

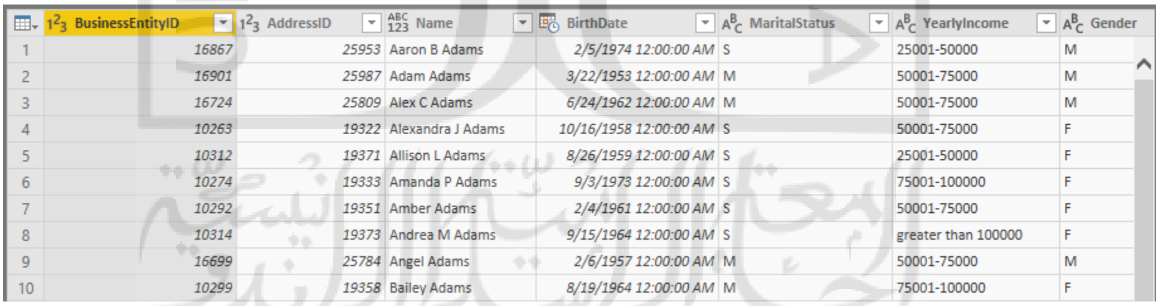
Figure 4.5 Table Result of PowerQuery DimProduct

2. DimCustomer



```
let
    Source = Sql.Database("RFFZNPC\SQLSERVER", "AdventureWorks2017"),
    Sales_vIndividualCustomer = Source{[Schema="Sales",Item="vIndividualCustomer"]}[Data],
    #"Added Custom" = Table.AddColumn(Sales_vIndividualCustomer, "Name", each if [MiddleName] = null then [FirstName]&" "&[LastName] else [FirstName]&" "&[MiddleName]&" "&[LastName]),
    #"Reordered Columns" = Table.ReorderColumns(#"Added Custom",{"BusinessEntityID", "Title", "Name", "FirstName", "MiddleName", "LastName", "Suffix", "PhoneNumber", "PhoneNumberType",
        "EmailAddress", "EmailPromotion", "AddressType", "AddressLine1", "AddressLine2", "City", "StateProvinceName", "PostalCode", "CountryRegionName", "Demographics"}),
    #"Removed Columns" = Table.RemoveColumns(#"Reordered Columns",{"FirstName", "MiddleName", "LastName"}),
    #"Removed Other Columns" = Table.SelectColumns(#"Removed Columns",{"BusinessEntityID", "Title", "Name", "PhoneNumber", "EmailAddress", "EmailPromotion", "AddressType", "AddressLine1",
        "City", "StateProvinceName", "PostalCode", "CountryRegionName"}),
    #"Merged Queries" = Table.NestedJoin(#"Removed Other Columns",{"BusinessEntityID"},#"Sales vPersonDemographics",{"BusinessEntityID"},"Sales vPersonDemographics",JoinKind.LeftOuter),
    #"Expanded Sales vPersonDemographics" = Table.ExpandTableColumn(#"Merged Queries", "Sales vPersonDemographics", {"BusinessEntityID", "TotalPurchaseYTD", "DateFirstPurchase",
        "BirthDate", "MaritalStatus", "YearlyIncome", "Gender", "TotalChildren", "NumberChildrenAtHome", "Education", "Occupation", "NumberCarsOwned"}, {"BusinessEntityID.1",
        "TotalPurchaseYTD", "DateFirstPurchase", "BirthDate", "MaritalStatus", "YearlyIncome", "Gender", "TotalChildren", "NumberChildrenAtHome", "Education", "Occupation",
        "NumberCarsOwned"}),
    #"Removed Columns1" = Table.RemoveColumns(#"Expanded Sales vPersonDemographics",{"BusinessEntityID.1", "TotalPurchaseYTD", "DateFirstPurchase"}),
    #"Merged Queries1" = Table.NestedJoin(#"Removed Columns1",{"BusinessEntityID"},BusinessEntityAddress,{"BusinessEntityID"},"Person BusinessEntityAddress",JoinKind.LeftOuter),
    #"Expanded Person BusinessEntityAddress" = Table.ExpandTableColumn(#"Merged Queries1", "Person BusinessEntityAddress", {"BusinessEntityID", "AddressID", "AddressTypeID",
        "Person.Address", "Person.AddressType"}, {"BusinessEntityID.1", "AddressID", "AddressTypeID", "Person.Address", "Person.AddressType"}),
    #"Removed Columns2" = Table.RemoveColumns(#"Expanded Person BusinessEntityAddress",{"BusinessEntityID.1"}),
    #"Reordered Columns1" = Table.ReorderColumns(#"Removed Columns2",{"BusinessEntityID", "AddressID", "Title", "Name", "PhoneNumber", "EmailAddress", "EmailPromotion", "AddressType",
        "AddressLine1", "City", "StateProvinceName", "PostalCode", "CountryRegionName", "BirthDate", "MaritalStatus", "YearlyIncome", "Gender", "TotalChildren", "NumberChildrenAtHome",
        "Education", "Occupation", "NumberCarsOwned", "AddressTypeID", "Person.Address", "Person.AddressType"}),
    #"Removed Other Columns1" = Table.SelectColumns(#"Reordered Columns1",{"BusinessEntityID", "AddressID", "Name", "BirthDate", "MaritalStatus", "YearlyIncome", "Gender", "TotalChildren",
        "NumberChildrenAtHome", "Education", "Occupation", "NumberCarsOwned"}),
    #"Removed Duplicates" = Table.Distinct(#"Removed Other Columns1", {"BusinessEntityID"}),
    #"Removed Columns3" = Table.RemoveColumns(#"Removed Duplicates",{"TotalChildren"})
in
    #"Removed Columns3"
```

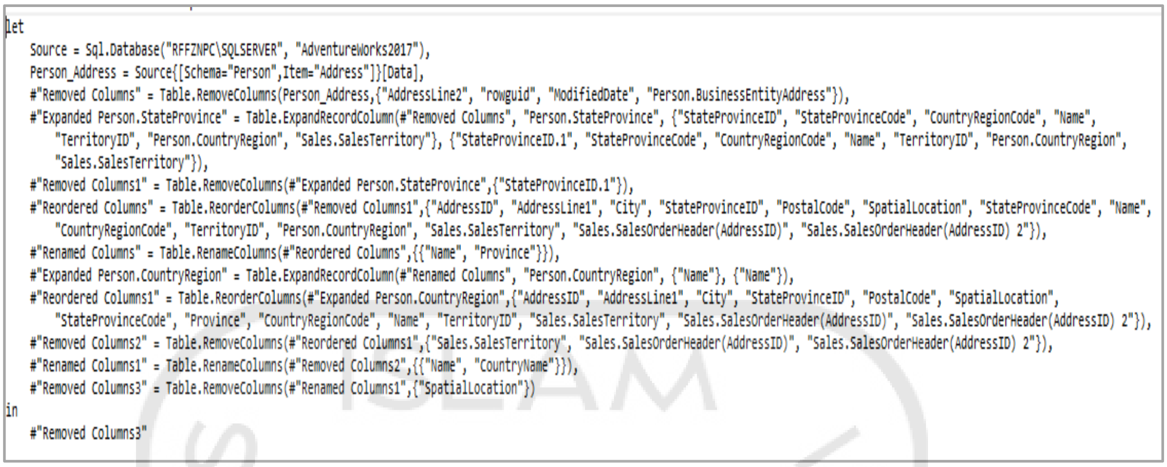
Figure 4.6 Pseudo Code of PowerQuery DimCustomer



	BusinessEntityID	AddressID	Name	BirthDate	MaritalStatus	YearlyIncome	Gender
1	16867	25953	Aaron B Adams	2/5/1974 12:00:00 AM	S	25001-50000	M
2	16901	25987	Adam Adams	3/22/1953 12:00:00 AM	M	50001-75000	M
3	16724	25809	Alex C Adams	6/24/1962 12:00:00 AM	M	50001-75000	M
4	10263	19322	Alexandra J Adams	10/16/1958 12:00:00 AM	S	50001-75000	F
5	10312	19371	Allison L Adams	8/26/1959 12:00:00 AM	S	25001-50000	F
6	10274	19333	Amanda P Adams	9/3/1973 12:00:00 AM	S	75001-100000	F
7	10292	19351	Amber Adams	2/4/1961 12:00:00 AM	S	50001-75000	F
8	10314	19373	Andrea M Adams	9/15/1964 12:00:00 AM	S	greater than 100000	F
9	16699	25784	Angel Adams	2/6/1957 12:00:00 AM	M	50001-75000	M
10	10299	19358	Bailey Adams	8/19/1964 12:00:00 AM	M	75001-100000	F

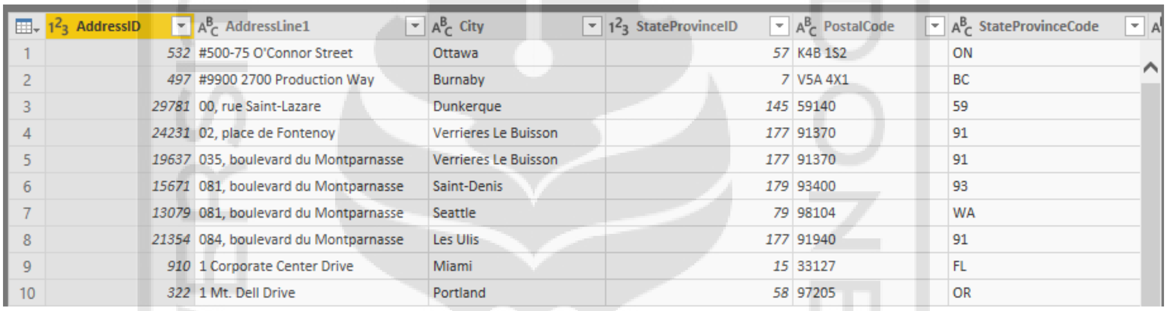
Figure 4.7 Table Result of PowerQuery DimCustomer

3. DimLocation



```
let
    Source = Sql.Database("RFFZNPC\SQLSERVER", "AdventureWorks2017"),
    Person_Address = Source{[Schema="Person",Item="Address"]}[Data],
    #"Removed Columns" = Table.RemoveColumns(Person_Address,{"AddressLine2", "rowguid", "ModifiedDate", "Person.BusinessEntityAddress"}),
    #"Expanded Person.StateProvince" = Table.ExpandRecordColumn(#"Removed Columns", "Person.StateProvince", {"StateProvinceID", "StateProvinceCode", "CountryRegionCode", "Name",
        "TerritoryID", "Person.CountryRegion", "Sales.SalesTerritory"}, {"StateProvinceID.1", "StateProvinceCode", "CountryRegionCode", "Name", "TerritoryID", "Person.CountryRegion",
        "Sales.SalesTerritory"}),
    #"Removed Columns1" = Table.RemoveColumns(#"Expanded Person.StateProvince",{"StateProvinceID.1"}),
    #"Reordered Columns" = Table.ReorderColumns(#"Removed Columns1",{"AddressID", "AddressLine1", "City", "StateProvinceID", "PostalCode", "SpatialLocation", "StateProvinceCode", "Name",
        "CountryRegionCode", "TerritoryID", "Person.CountryRegion", "Sales.SalesTerritory", "Sales.SalesOrderHeader(AddressID)", "Sales.SalesOrderHeader(AddressID) 2"}),
    #"Renamed Columns" = Table.RenameColumns(#"Reordered Columns",{{"Name", "Province"}}),
    #"Expanded Person.CountryRegion" = Table.ExpandRecordColumn(#"Renamed Columns", "Person.CountryRegion", {"Name"}, {"Name"}),
    #"Reordered Columns1" = Table.ReorderColumns(#"Expanded Person.CountryRegion",{"AddressID", "AddressLine1", "City", "StateProvinceID", "PostalCode", "SpatialLocation",
        "StateProvinceCode", "Province", "CountryRegionCode", "Name", "TerritoryID", "Sales.SalesTerritory", "Sales.SalesOrderHeader(AddressID)", "Sales.SalesOrderHeader(AddressID) 2"}),
    #"Removed Columns2" = Table.RemoveColumns(#"Reordered Columns1",{"Sales.SalesTerritory", "Sales.SalesOrderHeader(AddressID)", "Sales.SalesOrderHeader(AddressID) 2"}),
    #"Renamed Columns1" = Table.RenameColumns(#"Removed Columns2",{{"Name", "CountryName"}}),
    #"Removed Columns3" = Table.RemoveColumns(#"Renamed Columns1",{"SpatialLocation"})
in
    #"Removed Columns3"
```

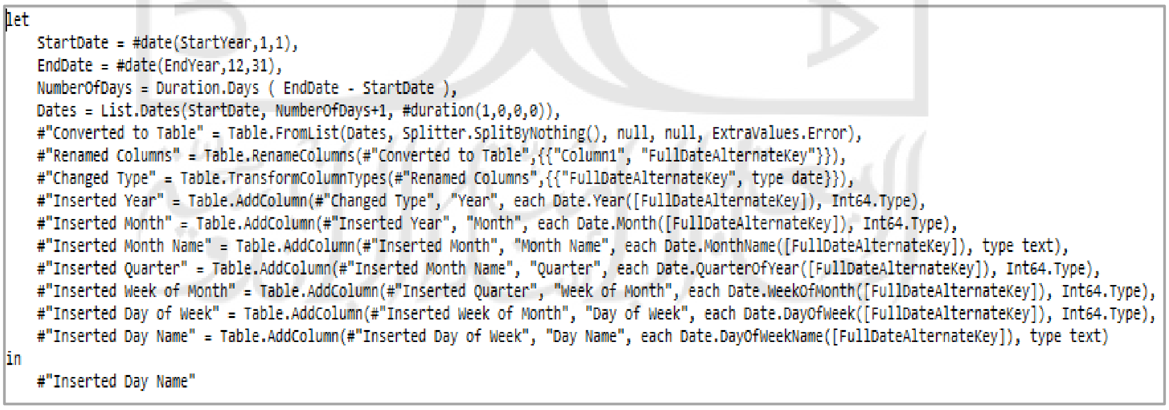
Figure 4.8 Pseudo Code of PowerQuery DimLocation



	AddressID	AddressLine1	City	StateProvinceID	PostalCode	StateProvinceCode
1	532	#500-75 O'Connor Street	Ottawa	57	K4B 1S2	ON
2	497	#9900 2700 Production Way	Burnaby	7	V5A 4X1	BC
3	29781	00, rue Saint-Lazare	Dunkerque	145	59140	59
4	24231	02, place de Fontenoy	Verrieres Le Buisson	177	91370	91
5	19637	035, boulevard du Montparnasse	Verrieres Le Buisson	177	91370	91
6	15671	081, boulevard du Montparnasse	Saint-Denis	179	93400	93
7	13079	081, boulevard du Montparnasse	Seattle	79	98104	WA
8	21354	084, boulevard du Montparnasse	Les Ulis	177	91940	91
9	910	1 Corporate Center Drive	Miami	15	33127	FL
10	322	1 Mt. Dell Drive	Portland	58	97205	OR

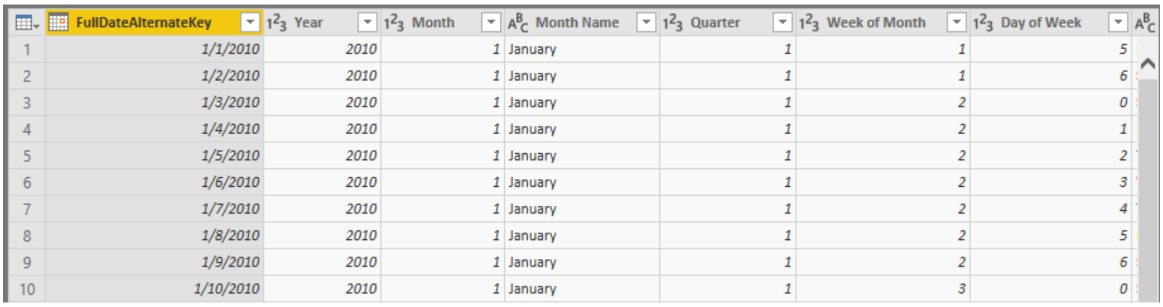
Figure 4.9 Table Result of PowerQuery DimLocation

4. DimDate



```
let
    StartDate = #date(StartYear,1,1),
    EndDate = #date(EndYear,12,31),
    NumberOfDays = Duration.Days ( EndDate - StartDate ),
    Dates = List.Dates(StartDate, NumberOfDays+1, #duration(1,0,0,0)),
    #"Converted to Table" = Table.FromList(Dates, Splitter.SplitByNothing(), null, null, ExtraValues.Error),
    #"Renamed Columns" = Table.RenameColumns(#"Converted to Table",{{"Column1", "FullDateAlternateKey"}}),
    #"Changed Type" = Table.TransformColumnTypes(#"Renamed Columns",{{"FullDateAlternateKey", type date}}),
    #"Inserted Year" = Table.AddColumn(#"Changed Type", "Year", each Date.Year([FullDateAlternateKey]), Int64.Type),
    #"Inserted Month" = Table.AddColumn(#"Inserted Year", "Month", each Date.Month([FullDateAlternateKey]), Int64.Type),
    #"Inserted Month Name" = Table.AddColumn(#"Inserted Month", "Month Name", each Date.MonthName([FullDateAlternateKey]), type text),
    #"Inserted Quarter" = Table.AddColumn(#"Inserted Month Name", "Quarter", each Date.QuarterOfYear([FullDateAlternateKey]), Int64.Type),
    #"Inserted Week of Month" = Table.AddColumn(#"Inserted Quarter", "Week of Month", each Date.WeekOfMonth([FullDateAlternateKey]), Int64.Type),
    #"Inserted Day of Week" = Table.AddColumn(#"Inserted Week of Month", "Day of Week", each Date.DayOfWeek([FullDateAlternateKey]), Int64.Type),
    #"Inserted Day Name" = Table.AddColumn(#"Inserted Day of Week", "Day Name", each Date.DayOfWeekName([FullDateAlternateKey]), type text)
in
    #"Inserted Day Name"
```

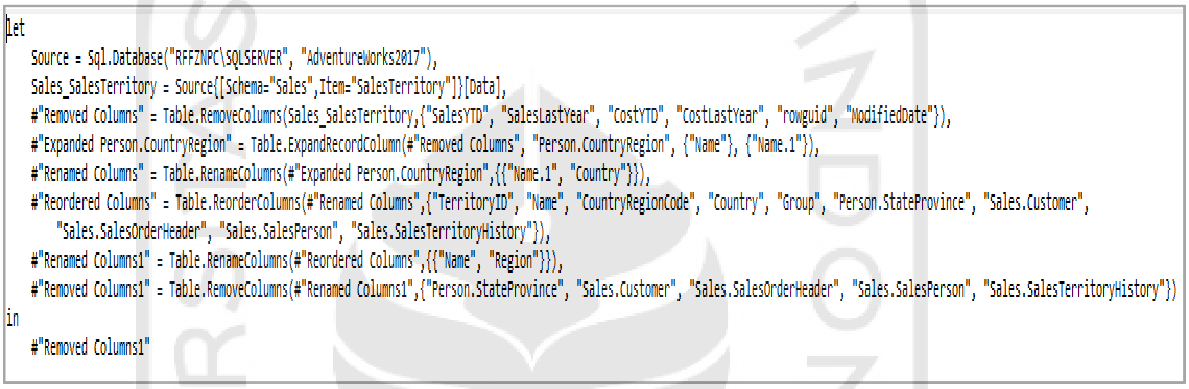
Figure 4.10 Pseudo Code of PowerQuery DimDate



	FullDateAlternateKey	Year	Month	Month Name	Quarter	Week of Month	Day of Week
1	1/1/2010	2010	1	January	1	1	5
2	1/2/2010	2010	1	January	1	1	6
3	1/3/2010	2010	1	January	1	2	0
4	1/4/2010	2010	1	January	1	2	1
5	1/5/2010	2010	1	January	1	2	2
6	1/6/2010	2010	1	January	1	2	3
7	1/7/2010	2010	1	January	1	2	4
8	1/8/2010	2010	1	January	1	2	5
9	1/9/2010	2010	1	January	1	2	6
10	1/10/2010	2010	1	January	1	3	0

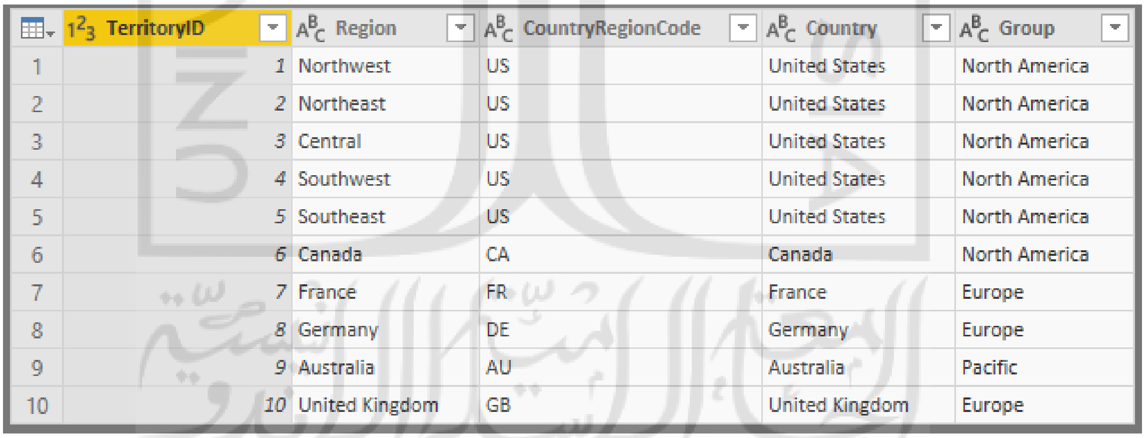
Figure 4.11 Table Result of PowerQuery DimDate

5. DimSalesTerritory



```
let
    Source = Sql.Database("RFFZNPC\SQLSERVER", "AdventureWorks2017"),
    Sales_SalesTerritory = Source{[Schema="Sales",Item="SalesTerritory"]}[Data],
    #"Removed Columns" = Table.RemoveColumns(Sales_SalesTerritory,{"SalesYTD", "SalesLastYear", "CostYTD", "CostLastYear", "rowguid", "ModifiedDate"}),
    #"Expanded Person.CountryRegion" = Table.ExpandRecordColumn(#"Removed Columns", "Person.CountryRegion", {"Name"}, {"Name.1"}),
    #"Renamed Columns" = Table.RenameColumns(#"Expanded Person.CountryRegion",{{"Name.1", "Country"}}),
    #"Reordered Columns" = Table.ReorderColumns(#"Renamed Columns",{"TerritoryID", "Name", "CountryRegionCode", "Country", "Group", "Person.StateProvince", "Sales.Customer",
        "Sales.SalesOrderHeader", "Sales.SalesPerson", "Sales.SalesTerritoryHistory"}),
    #"Renamed Columns1" = Table.RenameColumns(#"Reordered Columns",{{"Name", "Region"}}),
    #"Removed Columns1" = Table.RemoveColumns(#"Renamed Columns1",{"Person.StateProvince", "Sales.Customer", "Sales.SalesOrderHeader", "Sales.SalesPerson", "Sales.SalesTerritoryHistory"})
in
    #"Removed Columns1"
```

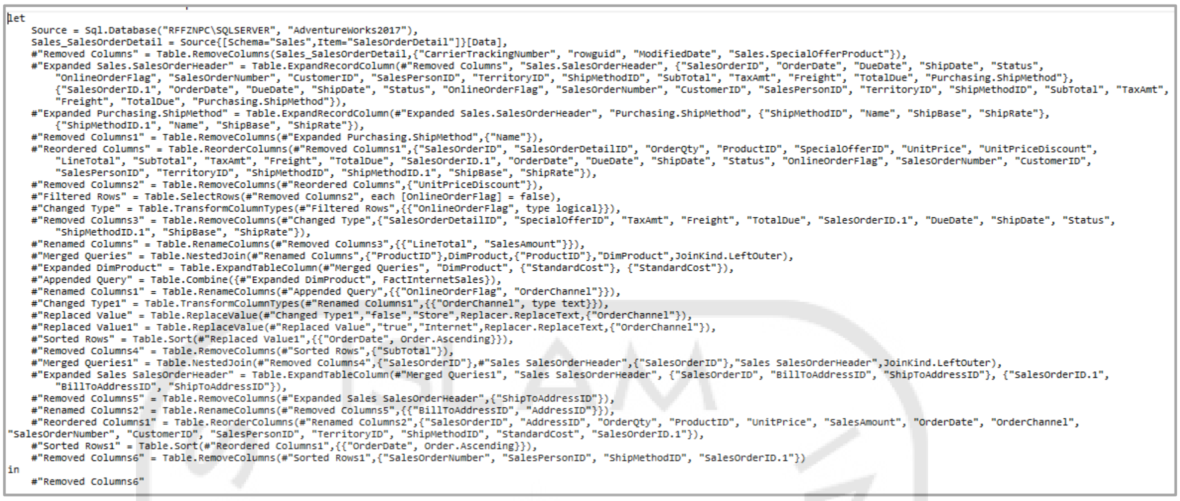
Figure 4.12 Pseudo Code of PowerQuery DimSalesTerritory



	TerritoryID	Region	CountryRegionCode	Country	Group
1	1	Northwest	US	United States	North America
2	2	Northeast	US	United States	North America
3	3	Central	US	United States	North America
4	4	Southwest	US	United States	North America
5	5	Southeast	US	United States	North America
6	6	Canada	CA	Canada	North America
7	7	France	FR	France	Europe
8	8	Germany	DE	Germany	Europe
9	9	Australia	AU	Australia	Pacific
10	10	United Kingdom	GB	United Kingdom	Europe

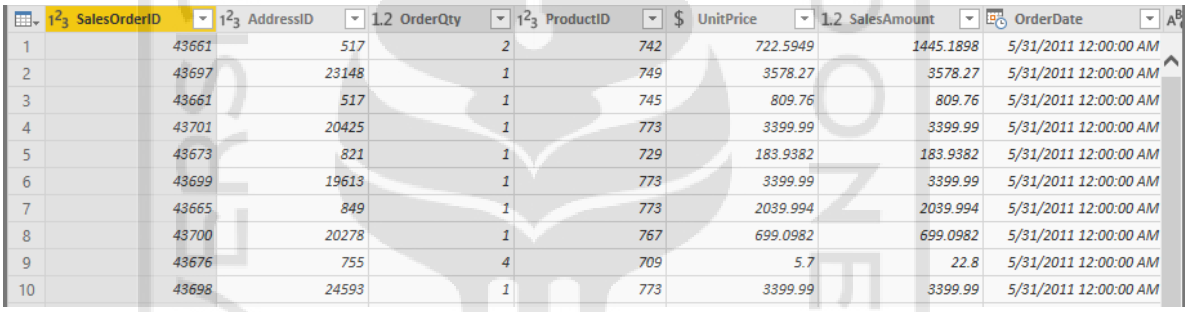
Figure 4.13 Table Result of PowerQuery DimSalesTerritory

6. DimFactTable



```
let
    Source = Sql.Database("RFFZNPC\SQLSERVER", "AdventureWorks2017"),
    Sales_SalesOrderDetail = Source{[Schema="Sales",Item="SalesOrderDetail"]}[Data],
    #"Removed Columns" = Table.RemoveColumns(Sales_SalesOrderDetail,{"CarrierTrackingNumber", "rowguid", "ModifiedDate", "Sales.SpecialOfferProduct"}),
    #"Expanded Sales.SalesOrderHeader" = Table.ExpandRecordColumn(#"Removed Columns", "Sales.SalesOrderHeader", {"SalesOrderID", "OrderDate", "DueDate", "ShipDate", "Status",
        "OnlineOrderFlag", "SalesOrderNumber", "CustomerID", "SalesPersonID", "TerritoryID", "ShipMethodID", "SubTotal", "TaxAmt", "Freight", "TotalDue", "Purchasing.ShipMethod"},
        {"SalesOrderID.1", "OrderDate", "DueDate", "ShipDate", "Status", "OnlineOrderFlag", "SalesOrderNumber", "CustomerID", "SalesPersonID", "TerritoryID", "ShipMethodID", "SubTotal", "TaxAmt",
        "Freight", "TotalDue", "Purchasing.ShipMethod"}),
    #"Expanded Purchasing.ShipMethod" = Table.ExpandRecordColumn(#"Expanded Sales.SalesOrderHeader", "Purchasing.ShipMethod", {"ShipMethodID", "Name", "ShipBase", "ShipRate"},
        {"ShipMethodID.1", "Name", "ShipBase", "ShipRate"}),
    #"Removed Columns1" = Table.RemoveColumns(#"Expanded Purchasing.ShipMethod",{"Name"}),
    #"Reordered Columns" = Table.ReorderColumns(#"Removed Columns1",{"SalesOrderID", "SalesOrderDetailID", "OrderQty", "ProductID", "SpecialOfferID", "UnitPrice", "UnitPriceDiscount",
        "LineTotal", "SubTotal", "TaxAmt", "Freight", "TotalDue", "SalesOrderID.1", "OrderDate", "DueDate", "ShipDate", "Status", "OnlineOrderFlag", "SalesOrderNumber", "CustomerID",
        "SalesPersonID", "TerritoryID", "ShipMethodID", "ShipMethodID.1", "ShipBase", "ShipRate"}),
    #"Removed Columns2" = Table.RemoveColumns(#"Reordered Columns",{"UnitPriceDiscount"}),
    #"Filtered Rows" = Table.SelectRows(#"Removed Columns2", each [OnlineOrderFlag] = false),
    #"Changed Type" = Table.TransformColumnTypes(#"Filtered Rows",{{"OnlineOrderFlag", type logical}}),
    #"Removed Columns3" = Table.RemoveColumns(#"Changed Type",{"SalesOrderDetailID", "SpecialOfferID", "TaxAmt", "Freight", "TotalDue", "SalesOrderID.1", "DueDate", "ShipDate", "Status",
        "ShipMethodID.1", "ShipBase", "ShipRate"}),
    #"Renamed Columns" = Table.RenameColumns(#"Removed Columns3",{{"LineTotal", "SalesAmount"}}),
    #"Merged Queries" = Table.NestedJoin(#"Renamed Columns",{"ProductID"},DimProduct,{"ProductID"},"DimProduct",JoinKind.LeftOuter),
    #"Expanded DimProduct" = Table.ExpandTableColumn(#"Merged Queries", "DimProduct", {"StandardCost"}, {"StandardCost"}),
    #"Appended Query" = Table.Combine({#"Expanded DimProduct", FactInternetSales}),
    #"Renamed Columns1" = Table.RenameColumns(#"Appended Query",{{"OnlineOrderFlag", "OrderChannel"}}),
    #"Changed Type1" = Table.TransformColumnTypes(#"Renamed Columns1",{{"OrderChannel", type text}}),
    #"Replaced Value" = Table.ReplaceValue(#"Changed Type1","false","Store",Replacer.ReplaceText,{"OrderChannel"}),
    #"Replaced Value1" = Table.ReplaceValue(#"Replaced Value","true","Internet",Replacer.ReplaceText,{"OrderChannel"}),
    #"Sorted Rows" = Table.Sort(#"Replaced Value1",{{"OrderDate", Order.Ascending}}),
    #"Removed Columns4" = Table.RemoveColumns(#"Sorted Rows",{"SubTotal"}),
    #"Merged Queries1" = Table.NestedJoin(#"Removed Columns4",{"SalesOrderID"},#"Sales SalesOrderHeader",{"SalesOrderID"},"Sales SalesOrderHeader",JoinKind.LeftOuter),
    #"Expanded Sales SalesOrderHeader" = Table.ExpandTableColumn(#"Merged Queries1", "Sales SalesOrderHeader", {"SalesOrderID", "BillToAddressID", "ShipToAddressID"}, {"SalesOrderID.1",
        "BillToAddressID", "ShipToAddressID"}),
    #"Removed Columns5" = Table.RemoveColumns(#"Expanded Sales SalesOrderHeader",{"ShipToAddressID"}),
    #"Renamed Columns2" = Table.RenameColumns(#"Removed Columns5",{{"BillToAddressID", "AddressID"}}),
    #"Reordered Columns1" = Table.ReorderColumns(#"Renamed Columns2",{"SalesOrderID", "AddressID", "OrderQty", "ProductID", "UnitPrice", "SalesAmount", "OrderDate", "OrderChannel",
"SalesOrderNumber", "CustomerID", "SalesPersonID", "TerritoryID", "ShipMethodID", "StandardCost", "SalesOrderID.1"}),
    #"Sorted Rows1" = Table.Sort(#"Reordered Columns1",{{"OrderDate", Order.Ascending}}),
    #"Removed Columns6" = Table.RemoveColumns(#"Sorted Rows1",{"SalesOrderNumber", "SalesPersonID", "ShipMethodID", "SalesOrderID.1"})
in
    #"Removed Columns6"
```

Figure 4.14 Pseudo Code of PowerQuery DimFactTable

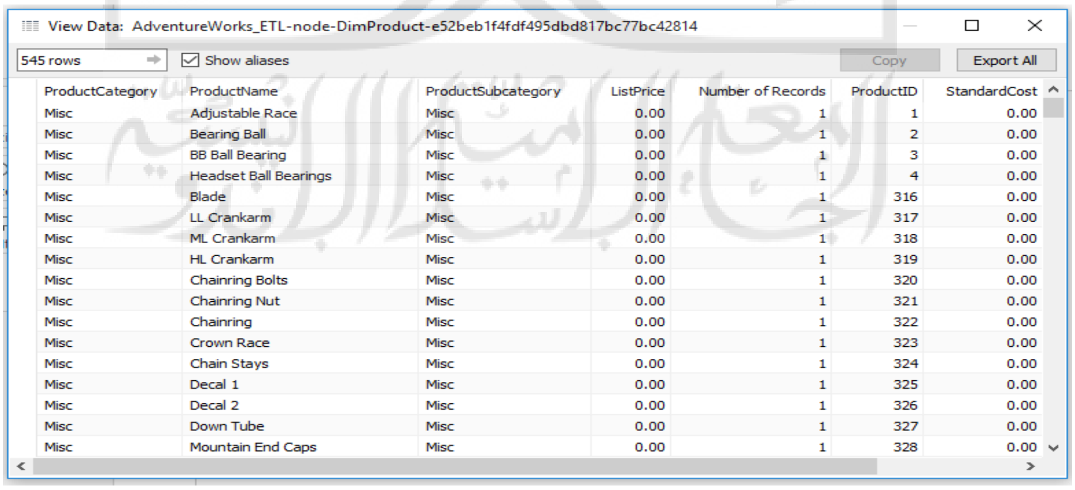


	SalesOrderID	AddressID	OrderQty	ProductID	UnitPrice	SalesAmount	OrderDate
1	43661	517	2	742	722.5949	1445.1898	5/31/2011 12:00:00 AM
2	43697	23148	1	749	3578.27	3578.27	5/31/2011 12:00:00 AM
3	43661	517	1	745	809.76	809.76	5/31/2011 12:00:00 AM
4	43701	20425	1	773	3399.99	3399.99	5/31/2011 12:00:00 AM
5	43673	821	1	729	183.9382	183.9382	5/31/2011 12:00:00 AM
6	43699	19613	1	773	3399.99	3399.99	5/31/2011 12:00:00 AM
7	43665	849	1	773	2039.994	2039.994	5/31/2011 12:00:00 AM
8	43700	20278	1	767	699.0982	699.0982	5/31/2011 12:00:00 AM
9	43676	755	4	709	5.7	22.8	5/31/2011 12:00:00 AM
10	43698	24593	1	773	3399.99	3399.99	5/31/2011 12:00:00 AM

Figure 4.15 Table Result of PowerQuery DimFactTable

4.2.3.2 Tableau Result

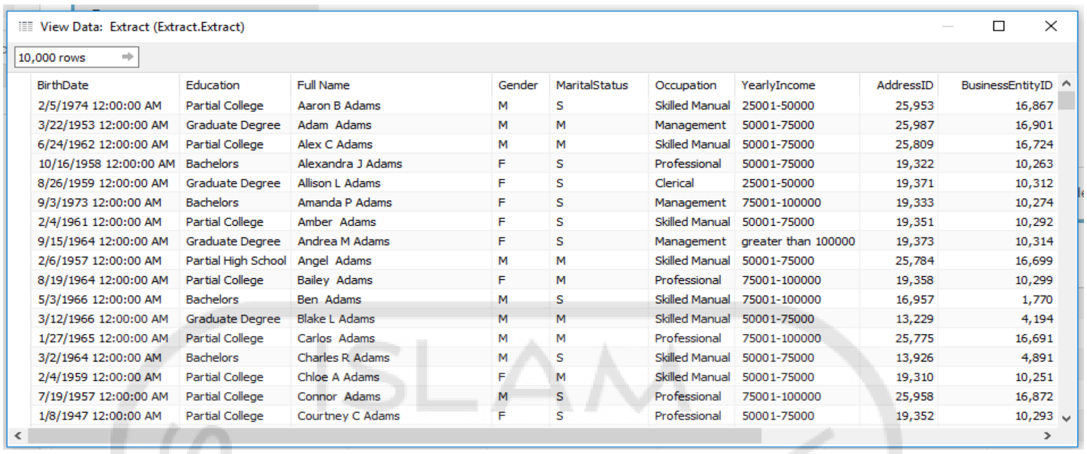
1. DimProduct


View Data: AdventureWorks_ETL-node-DimProduct-e52beb1f4fdf495dbd817bc77bc42814
545 rows
☑ Show aliases
Copy
Export All

ProductCategory	ProductName	ProductSubcategory	ListPrice	Number of Records	ProductID	StandardCost
Misc	Adjustable Race	Misc	0.00	1	1	0.00
Misc	Bearing Ball	Misc	0.00	1	2	0.00
Misc	BB Ball Bearing	Misc	0.00	1	3	0.00
Misc	Headset Ball Bearings	Misc	0.00	1	4	0.00
Misc	Blade	Misc	0.00	1	316	0.00
Misc	LL Crankarm	Misc	0.00	1	317	0.00
Misc	ML Crankarm	Misc	0.00	1	318	0.00
Misc	HL Crankarm	Misc	0.00	1	319	0.00
Misc	Chainring Bolts	Misc	0.00	1	320	0.00
Misc	Chainring Nut	Misc	0.00	1	321	0.00
Misc	Chainring	Misc	0.00	1	322	0.00
Misc	Crown Race	Misc	0.00	1	323	0.00
Misc	Chain Stays	Misc	0.00	1	324	0.00
Misc	Decal 1	Misc	0.00	1	325	0.00
Misc	Decal 2	Misc	0.00	1	326	0.00
Misc	Down Tube	Misc	0.00	1	327	0.00
Misc	Mountain End Caps	Misc	0.00	1	328	0.00

Figure 4.16 Table Result of Tableau Prep Builder DimProduct

2. DimCustomer



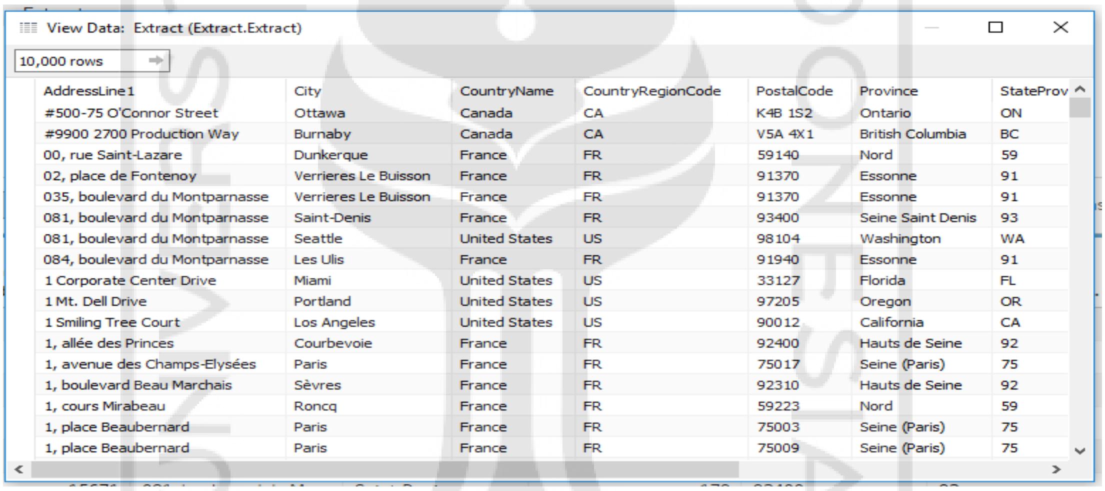
View Data: Extract (Extract.Extract)

10,000 rows

BirthDate	Education	Full Name	Gender	MaritalStatus	Occupation	YearlyIncome	AddressID	BusinessEntityID
2/5/1974 12:00:00 AM	Partial College	Aaron B Adams	M	S	Skilled Manual	25001-50000	25,953	16,867
3/22/1953 12:00:00 AM	Graduate Degree	Adam Adams	M	M	Management	50001-75000	25,987	16,901
6/24/1962 12:00:00 AM	Partial College	Alex C Adams	M	M	Skilled Manual	50001-75000	25,809	16,724
10/16/1958 12:00:00 AM	Bachelors	Alexandra J Adams	F	S	Professional	50001-75000	19,322	10,263
8/26/1959 12:00:00 AM	Graduate Degree	Allison L Adams	F	S	Clerical	25001-50000	19,371	10,312
9/3/1973 12:00:00 AM	Bachelors	Amanda P Adams	F	S	Management	75001-100000	19,333	10,274
2/4/1961 12:00:00 AM	Partial College	Amber Adams	F	S	Skilled Manual	50001-75000	19,351	10,292
9/15/1964 12:00:00 AM	Graduate Degree	Andrea M Adams	F	S	Management	greater than 100000	19,373	10,314
2/6/1957 12:00:00 AM	Partial High School	Angel Adams	M	M	Skilled Manual	50001-75000	25,784	16,699
8/19/1964 12:00:00 AM	Partial College	Bailey Adams	F	M	Professional	75001-100000	19,358	10,299
5/3/1966 12:00:00 AM	Bachelors	Ben Adams	M	S	Skilled Manual	75001-100000	16,957	1,770
3/12/1966 12:00:00 AM	Graduate Degree	Blake L Adams	M	M	Skilled Manual	50001-75000	13,229	4,194
1/27/1965 12:00:00 AM	Partial College	Carlos Adams	M	M	Professional	75001-100000	25,775	16,691
3/2/1964 12:00:00 AM	Bachelors	Charles R Adams	M	S	Skilled Manual	50001-75000	13,926	4,891
2/4/1959 12:00:00 AM	Partial College	Chloe A Adams	F	M	Skilled Manual	50001-75000	19,310	10,251
7/19/1957 12:00:00 AM	Partial College	Connor Adams	M	S	Professional	75001-100000	25,958	16,872
1/8/1947 12:00:00 AM	Partial College	Courtney C Adams	F	S	Professional	50001-75000	19,352	10,293

Figure 4.17 Table Result of Tableau Prep Builder DimCustomer

3. DimLocation



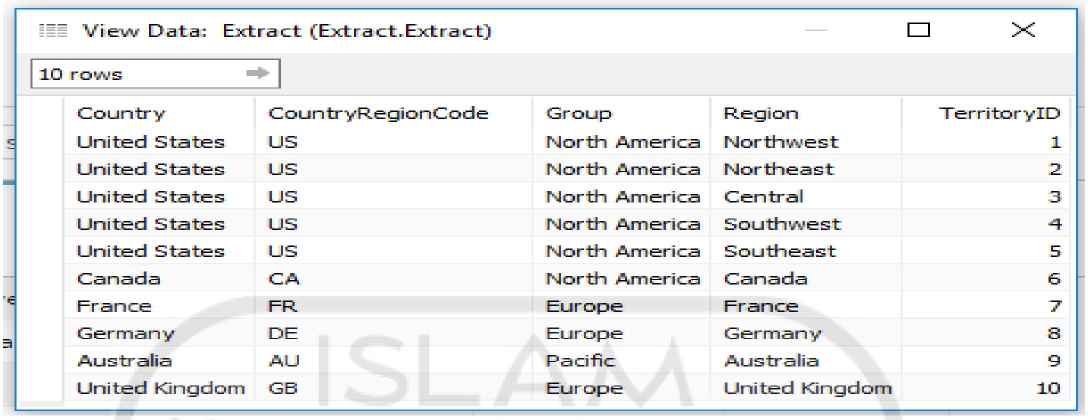
View Data: Extract (Extract.Extract)

10,000 rows

AddressLine1	City	CountryName	CountryRegionCode	PostalCode	Province	StateProv
#500-75 O'Connor Street	Ottawa	Canada	CA	K4B 1S2	Ontario	ON
#9900 2700 Production Way	Burnaby	Canada	CA	V5A 4X1	British Columbia	BC
00, rue Saint-Lazare	Dunkerque	France	FR	59140	Nord	59
02, place de Fontenoy	Verrieres Le Buisson	France	FR	91370	Essonne	91
035, boulevard du Montparnasse	Verrieres Le Buisson	France	FR	91370	Essonne	91
081, boulevard du Montparnasse	Saint-Denis	France	FR	93400	Seine Saint Denis	93
081, boulevard du Montparnasse	Seattle	United States	US	98104	Washington	WA
084, boulevard du Montparnasse	Les Ulis	France	FR	91940	Essonne	91
1 Corporate Center Drive	Miami	United States	US	33127	Florida	FL
1 Mt. Dell Drive	Portland	United States	US	97205	Oregon	OR
1 Smiling Tree Court	Los Angeles	United States	US	90012	California	CA
1, allée des Princes	Courbevoie	France	FR	92400	Hauts de Seine	92
1, avenue des Champs-Elysées	Paris	France	FR	75017	Seine (Paris)	75
1, boulevard Beau Marchais	Sèvres	France	FR	92310	Hauts de Seine	92
1, cours Mirabeau	Roncq	France	FR	59223	Nord	59
1, place Beaubernard	Paris	France	FR	75003	Seine (Paris)	75
1, place Beaubernard	Paris	France	FR	75009	Seine (Paris)	75

Figure 4.18 Table Result of Tableau Prep Builder DimLocation

4. DimSalesTerritory



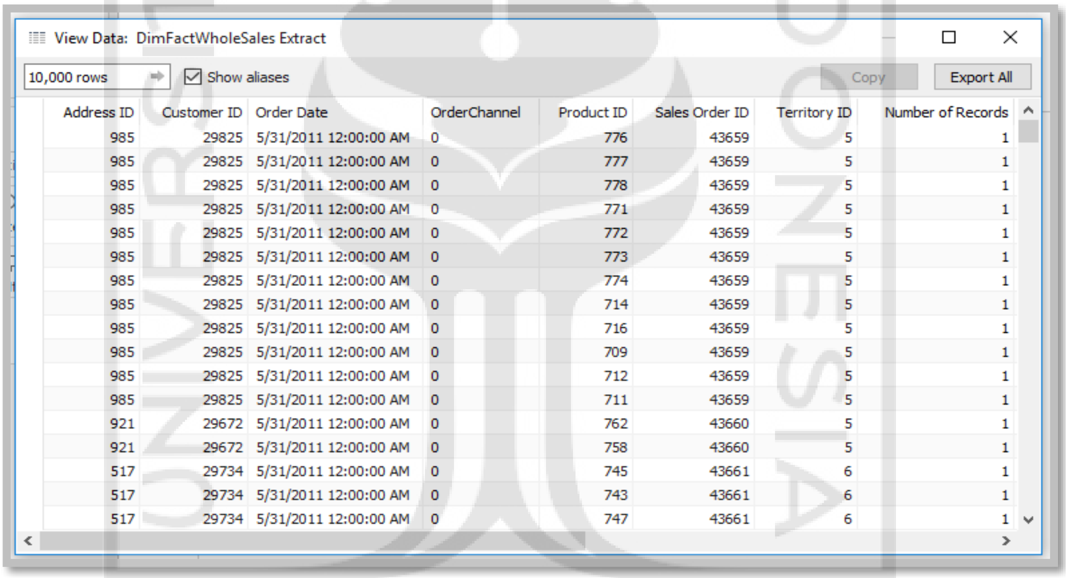
View Data: Extract (Extract.Extract)

10 rows

Country	CountryRegionCode	Group	Region	TerritoryID
United States	US	North America	Northwest	1
United States	US	North America	Northeast	2
United States	US	North America	Central	3
United States	US	North America	Southwest	4
United States	US	North America	Southeast	5
Canada	CA	North America	Canada	6
France	FR	Europe	France	7
Germany	DE	Europe	Germany	8
Australia	AU	Pacific	Australia	9
United Kingdom	GB	Europe	United Kingdom	10

Figure 4.19 Table Result of Tableau Prep Builder DimSalesTerritory

5. DimFactTable



View Data: DimFactWholeSales Extract

10,000 rows ☑ Show aliases

Address ID	Customer ID	Order Date	OrderChannel	Product ID	Sales Order ID	Territory ID	Number of Records
985	29825	5/31/2011 12:00:00 AM	0	776	43659	5	1
985	29825	5/31/2011 12:00:00 AM	0	777	43659	5	1
985	29825	5/31/2011 12:00:00 AM	0	778	43659	5	1
985	29825	5/31/2011 12:00:00 AM	0	771	43659	5	1
985	29825	5/31/2011 12:00:00 AM	0	772	43659	5	1
985	29825	5/31/2011 12:00:00 AM	0	773	43659	5	1
985	29825	5/31/2011 12:00:00 AM	0	774	43659	5	1
985	29825	5/31/2011 12:00:00 AM	0	714	43659	5	1
985	29825	5/31/2011 12:00:00 AM	0	716	43659	5	1
985	29825	5/31/2011 12:00:00 AM	0	709	43659	5	1
985	29825	5/31/2011 12:00:00 AM	0	712	43659	5	1
985	29825	5/31/2011 12:00:00 AM	0	711	43659	5	1
921	29672	5/31/2011 12:00:00 AM	0	762	43660	5	1
921	29672	5/31/2011 12:00:00 AM	0	758	43660	5	1
517	29734	5/31/2011 12:00:00 AM	0	745	43661	6	1
517	29734	5/31/2011 12:00:00 AM	0	743	43661	6	1
517	29734	5/31/2011 12:00:00 AM	0	747	43661	6	1

Figure 4.20 Table Result of Tableau Prep Builder DimFactTable

4.2.4 BI Implementation

In this section, the implementation of BI using both software is the same, which is creating the dashboard for BI analysis. The dashboard is simplifying the report of BI implementation with a powerful and very good interface. The dashboard was developed by creating the calculation measurement that depends on the user’s goal. In this study, the researcher has created a simple dashboard which was Product Sales Analysis. This dashboard includes several DAX functions to create a measure, such as:

1. Sales Revenue: SUM(FactWholeSales[SalesAmount])
2. Sales Volume: SUM(FactWholeSales[OrderQty])

This is the result that developed using the dashboard in both software even in Power BI or Tableau Desktop. The dashboard is showed in figure 4.20 and 4.21:

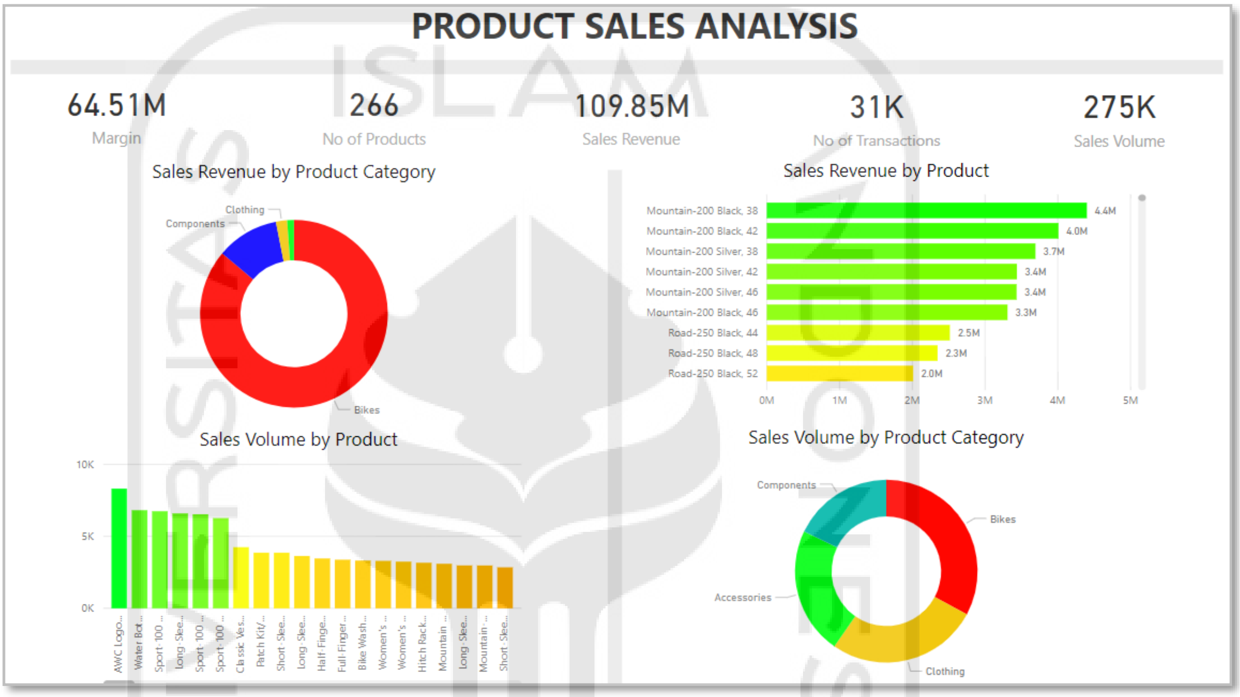


Figure 4.21 Result Dashboard in Power BI

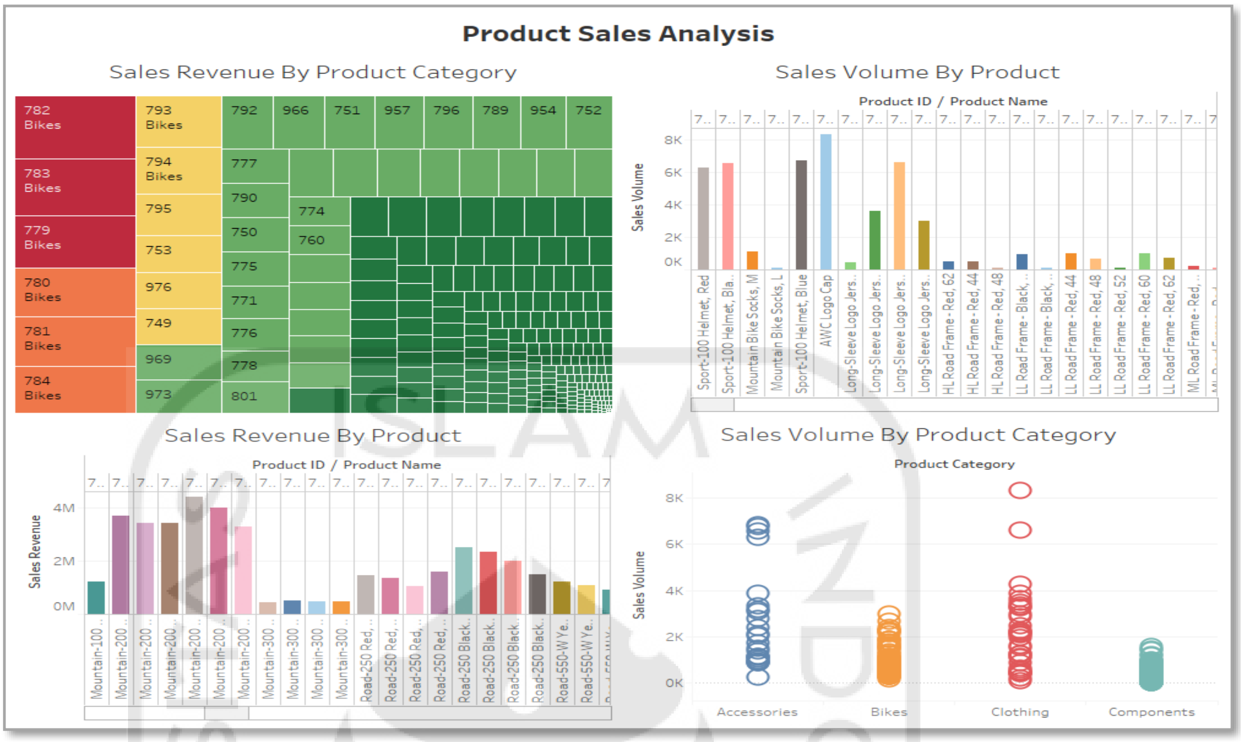


Figure 4.22 Result Dashboard in Tableau Desktop

4.2.5 Parameter Analysis

This section is the main segment of this research. The researcher conducted a deep analysis of comparative study between two most-famous SSBI Tools which were Power BI that include Power Query and Tableau Desktop that supported by Tableau Prep Builder. The comparison study using apple to apple comparison will be biased without parameter as the user satisfaction.

The parameter that the researcher already explained above in section 2.2.6, will be used to compare both software. Yet, to make the analysis is a focus and narrow, the researcher will limit the parameter based on the must-have features in SSBI Tools and more features that often to be the differentiators when selecting the software. The features are:

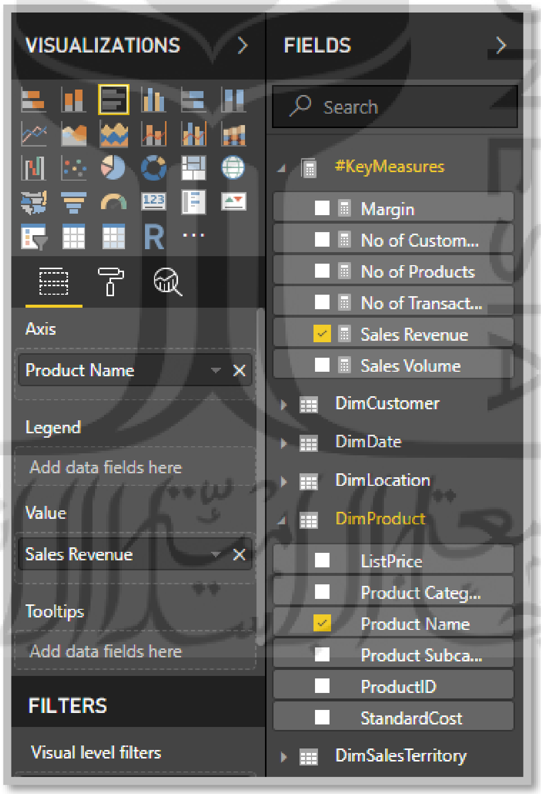
4.2.5.1 Must-Have SSBI Features

These features are unique because they provide more data management functionality for the business person creating an analytical application than for the information to the consumer who is relying on prebuilt BI applications with prebuilt integrated data. These features include:

1. Select data for analysis.

BI tools must enable the user to select the data used in decision-making analysis and present it as a pivot table-style interface where dimension attributes are placed in rows and columns, measures are selected, and filters are applied.

This feature found in both Power BI and Tableau Desktop, which Tableau Desktop is specifically designed where the dimension attributes are placed in rows and columns. There is also measures and filters box to apply. The feature captured in figure 4.23.



Axis
Product Name

Legend
Add data fields here

Value
Sales Revenue

Tooltips
Add data fields here

Figure 4.23 Power BI Interface of Select Data Analysis

Figure 4.23 shows the interface of Power BI feature of Select Data Analysis. The figure shows the dimension attributes chosen. The axis and the value in the figure above mean the place of attributes in rows and columns. There are also filters box in the "Tooltips" field.

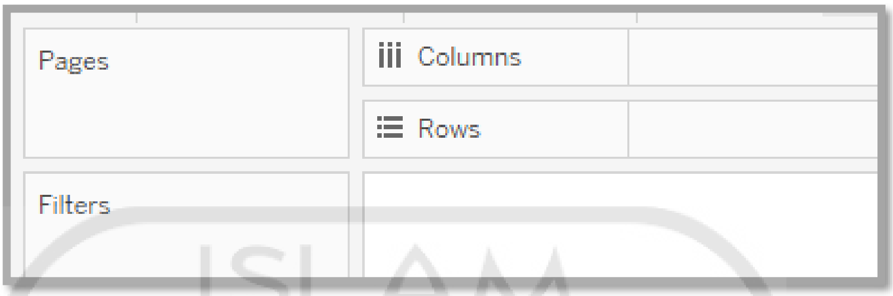


Figure 4.24 Tableau Desktop Interface of Select Data Analysis

Figure 4.24 shows the interface of Tableau Desktop feature of Select Data Analysis. The figure shows the dimension attributes which chosen and will be shown in the field. The figure shows the columns and rows field to be analyzed. It also shows the filters box for the additional parameter.

2. Data Blending

The product must permit the user to blend data from various data sources. This includes accessing the data and mapping or creating relationships with data from multiple sources.

This feature also found in each supported software both in Power BI also in Tableau Desktop. In Power BI, this feature is found in Power Query which can blend or combine the data from various data sources. In Tableau Desktop, this feature is found in Tableau Prep Builder which also can conduct this feature from various data sources. The feature is captured in figure 4.25 to 4.28.

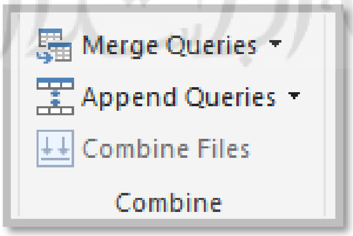


Figure 4.25 Power Query Blend Data Tools

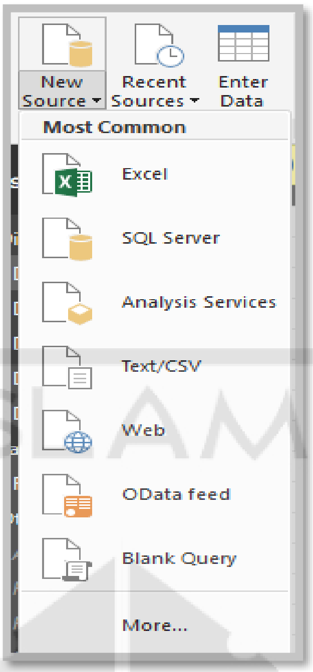


Figure 4.26 Power Query Data Sources

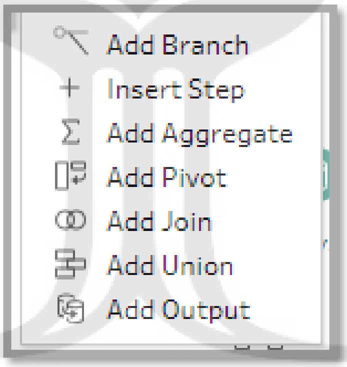
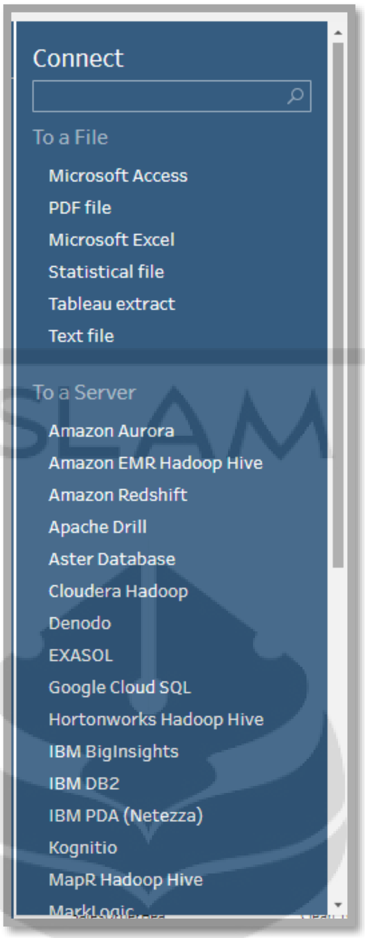


Figure 4.27 Tableau Prep Builder Blend Data Tools



Connect

To a File

Microsoft Access
PDF file
Microsoft Excel
Statistical file
Tableau extract
Text file

To a Server

Amazon Aurora
Amazon EMR Hadoop Hive
Amazon Redshift
Apache Drill
Aster Database
Cloudera Hadoop
Denodo
EXASOL
Google Cloud SQL
Hortonworks Hadoop Hive
IBM BigInsights
IBM DB2
IBM PDA (Netezza)
Kognitio
MapR Hadoop Hive
MarkLogic

Figure 4.28 Tableau Prep Builder Data Sources

3. Create measures

The product must enable the user to create and save measures or calculations for use in the analysis. These are also referred to as performance measures or key performance indicators.

This feature is also found in both Power BI and Tableau Desktop. In Power BI, this feature is created in Power BI Dashboard. While, in Tableau Desktop, this feature created in Tableau Desktop’s step analysis which is called as a worksheet. The figures 4.29 and 4.30 capture the feature in software:

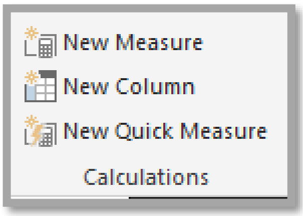


Figure 4.29 Power BI Measure Tools

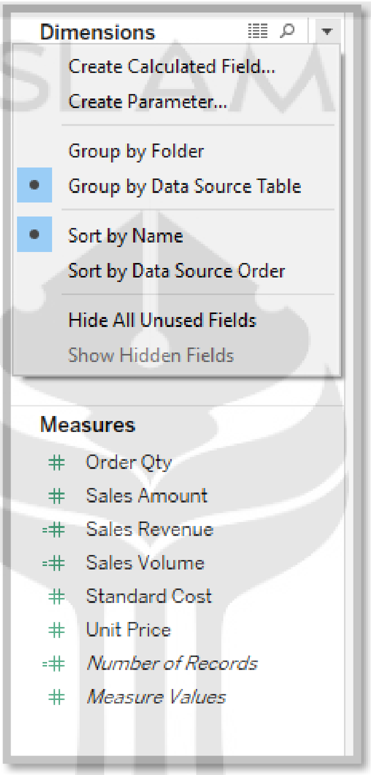


Figure 4.30 Tableau Desktop Measure Tools

In Figure 4.29, the measure will be created in the “create calculated field”. The calculated field can create various measures. The measures that already and directly created will be shown in the measures pane below the dimensions pane.

4. Create Hierarchies

The product must enable the user to create dimensional hierarchies, such as by geography or product, to a group and summarize data. This establishes the drill-down paths.

Both Power BI and Tableau Desktop own this feature. Both software can conduct drill-down hierarchy. In Power BI, the hierarchy created in fields pane directly on Dashboard Interface. In Tableau Desktop, the hierarchy created in dimensions pane on worksheet step before developing the dashboard interface. The Figures 4.31 and 4.32 capture the feature in software:

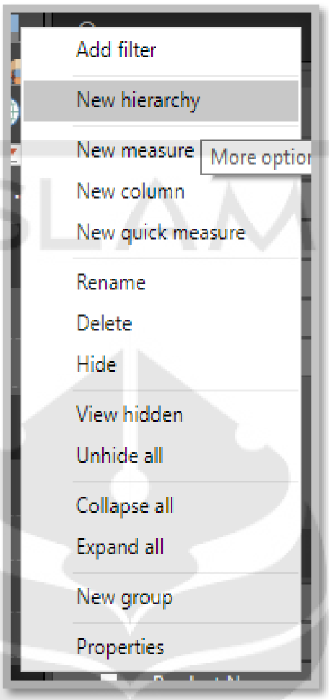


Figure 4.31 Power BI Drill-Down Hierarchy Tools

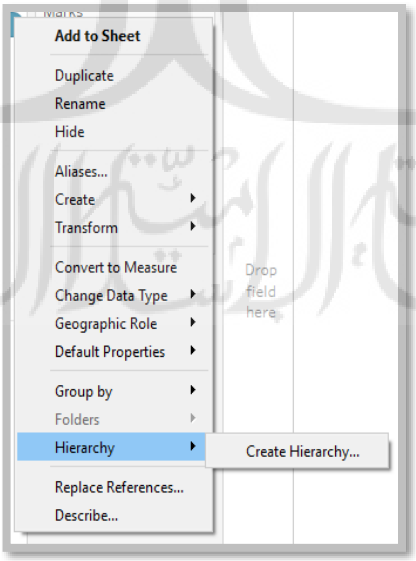


Figure 4.32 Tableau Desktop Drill-Down Hierarchy Tools

5. Save queries and analysis

The product should enable the BI user to save the data filters, selections and drill-down paths used in decision-making analysis so they can be reused.

This feature is found in both software. It is because both software conduct the flow transform using queries and the queries can be saved and reused. Both Power BI and Tableau Desktop can save the queries but the special extension to their own software. The figures 4.33 and 4.34 capture the feature in each software:

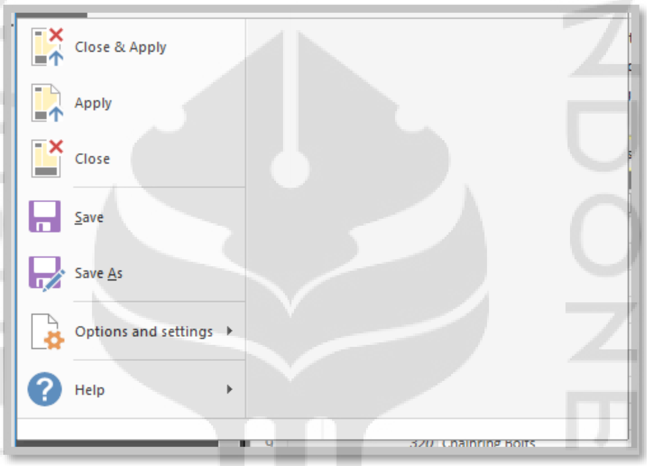


Figure 4.33 Power Query Save-Query Tools

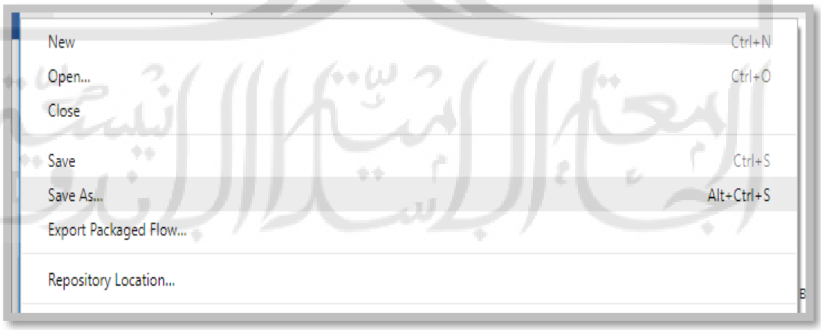


Figure 4.34 Tableau Prep Builder Save-Query Tools