Kelompok 8

- Atta A. Kurnia (2006836)
- Azzahra Alika (2105899)
- Azzam M. Naufal ()
- Muhamad N. Daffa (2003217)
- Riska Nurohmah (2109103)

In [1]:
```
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
#load dataset atau file csv ke DataFrame
df2 = pd.read_csv("C:\\Users\\riska\\Downloads\\data_returns\\orders_train.txt", sep=";")
```

In [3]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 481092 entries, 0 to 481091
Data columns (total 14 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   orderItemID     481092 non-null  int64
 1   orderDate       481092 non-null  object
 2   deliveryDate    481092 non-null  object
 3   itemID          481092 non-null  int64
 4   size            481092 non-null  object
 5   color           481092 non-null  object
 6   manufacturerID  481092 non-null  int64
 7   price           481092 non-null  float64
 8   customerID      481092 non-null  int64
 9   salutation      481092 non-null  object
 10  dateOfBirth     481092 non-null  object
 11  state           481092 non-null  object
 12  creationDate    481092 non-null  object
 13  returnShipment  481092 non-null  int64
dtypes: float64(1), int64(5), object(8)
memory usage: 51.4+ MB
```

In [4]:
```
df2.head(3)
```

Out[4]:

	orderItemID	orderDate	deliveryDate	itemID	size	color	manufacturerID	price	customerID	salutation	dateOfBirth	state	crea
0	1	2012-04-01	2012-04-03	186	m	denim	25	69.90	794	Mrs	1965-01-06	Baden-Wuerttemberg	20
1	2	2012-04-01	2012-04-03	71	9+	ocher	21	69.95	794	Mrs	1965-01-06	Baden-Wuerttemberg	20
2	3	2012-04-01	2012-04-03	71	9+	curry	21	69.95	794	Mrs	1965-01-06	Baden-Wuerttemberg	20

Terdapat datatype yang seharusnya merupakan datetime namun bertipe object, sehingga perlu diubah.

In [5]:
```
df2.isna().sum()
```

Out[5]:
```
orderItemID       0
orderDate         0
deliveryDate      0
itemID            0
size              0
color             0
manufacturerID    0
price             0
customerID        0
salutation        0
dateOfBirth       0
state             0
creationDate      0
returnShipment    0
dtype: int64
```

In [6]:
```
df2.isin(['?']).sum()
```

Out[6]:
```
orderItemID           0
orderDate             0
deliveryDate      39419
itemID                0
size                  0
color               143
manufacturerID        0
price                 0
customerID            0
salutation            0
dateOfBirth       48889
state                 0
creationDate          0
returnShipment        0
dtype: int64
```

praproses orderDate & deliveryDate

In [7]:
```
#melihat banyak data deliveryDate yang tahunnya 1990
df2['deliveryDate'].str.contains("1990").sum()
```

Out[7]: 4660

In [8]:
```
#melihat banyak data dateOfBirth yang tahunnya 1655
df2['dateOfBirth'].str.contains("1655").sum()
```

Out[8]: 3

Terdapat data yang tidak valid pada attribut beberapa attribut berupa '?' dan datetime dengan nilai tahun yang tidak masuk akal seperti '1990' dan '1655'

In [9]:
```
#mengubah entry invalid menjadi NaN
df2 = df2.replace('?', np.nan)
df2 = df2.replace('1990-12-31', np.nan)
df2 = df2.replace('1655-04-19', np.nan)
```

In [10]:
```
#mengubah tipe data attribut yang berupa tanggal menjadi datetime
df2['orderDate'] = pd.to_datetime(df2['orderDate'])
df2['deliveryDate'] = pd.to_datetime(df2['deliveryDate'])
df2['dateOfBirth'] = pd.to_datetime(df2['dateOfBirth'])
df2['creationDate'] = pd.to_datetime(df2['creationDate'])
```

In [11]:
```
#membuat attribut baru dari orderDate dan deliveryDate agar dapat direpresentasikan sebagai numerik
df2.insert(2, "orderYear", np.nan)
df2.insert(3, "orderMonth", np.nan)
df2.insert(4, "orderDated", np.nan)
df2.insert(6, "deliveryYear", np.nan)
df2.insert(7, "deliveryMonth", np.nan)
df2.insert(8, "deliveryDated", np.nan)
#membuat attribut baru baru berupa durasi delivery (dalam hari)
df2.insert(9, "deliveryDuration", np.nan)
```

In [12]:
```
#proses mengisi tanggal order yang sudah dipisah
from datetime import timedelta
df2["orderYear"] = df2["orderDate"].dt.year
df2["orderMonth"] = df2["orderDate"].dt.month
df2["orderDated"] = df2["orderDate"].dt.day
#proses mengisi deliveryDuration untuk baris yang deliveryDate dan orderDatenya bukan NaN
df2.loc[(df2["deliveryDate"].notna()), "deliveryDuration"] = (df2["deliveryDate"]- df2["orderDate"]) / np.timed
#proses mengisi deliveryDuration yang NaN dengan median yang tidak NaN
median_duration = df2["deliveryDuration"].median()
df2.loc[(df2["deliveryDate"].isna()), "deliveryDuration"] = median_duration

df2.loc[(df2["deliveryDate"].isna()), "deliveryDate"] = df2["orderDate"].loc[(df2["deliveryDate"].isna())] + ti
#proses mengisi tanggal delivery yang sudah dipisah
df2["deliveryYear"] = df2["deliveryDate"].dt.year
df2["deliveryMonth"] = df2["deliveryDate"].dt.month
df2["deliveryDated"] = df2["deliveryDate"].dt.day
```

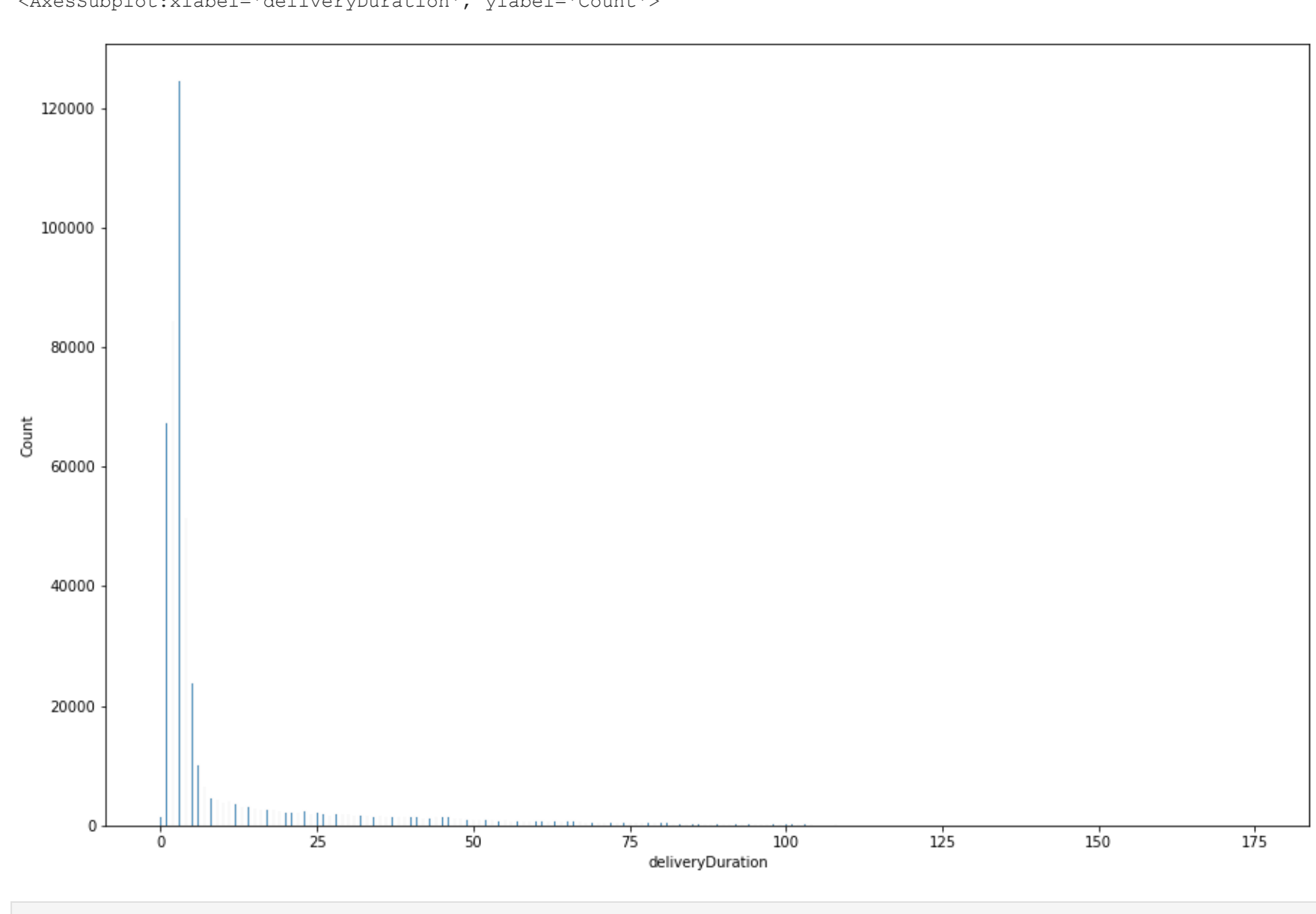
In [13]:
```
df2["orderYear"] = df2["orderYear"].astype('float')
df2["orderMonth"] = df2["orderMonth"].astype('float')
df2["orderDated"] = df2["orderDated"].astype('float')

df2["deliveryYear"] = df2["deliveryYear"].astype('float')
df2["deliveryMonth"] = df2["deliveryMonth"].astype('float')
df2["deliveryDated"] = df2["deliveryDated"].astype('float')

df2["deliveryDuration"] = df2["deliveryDuration"].astype('float')
```

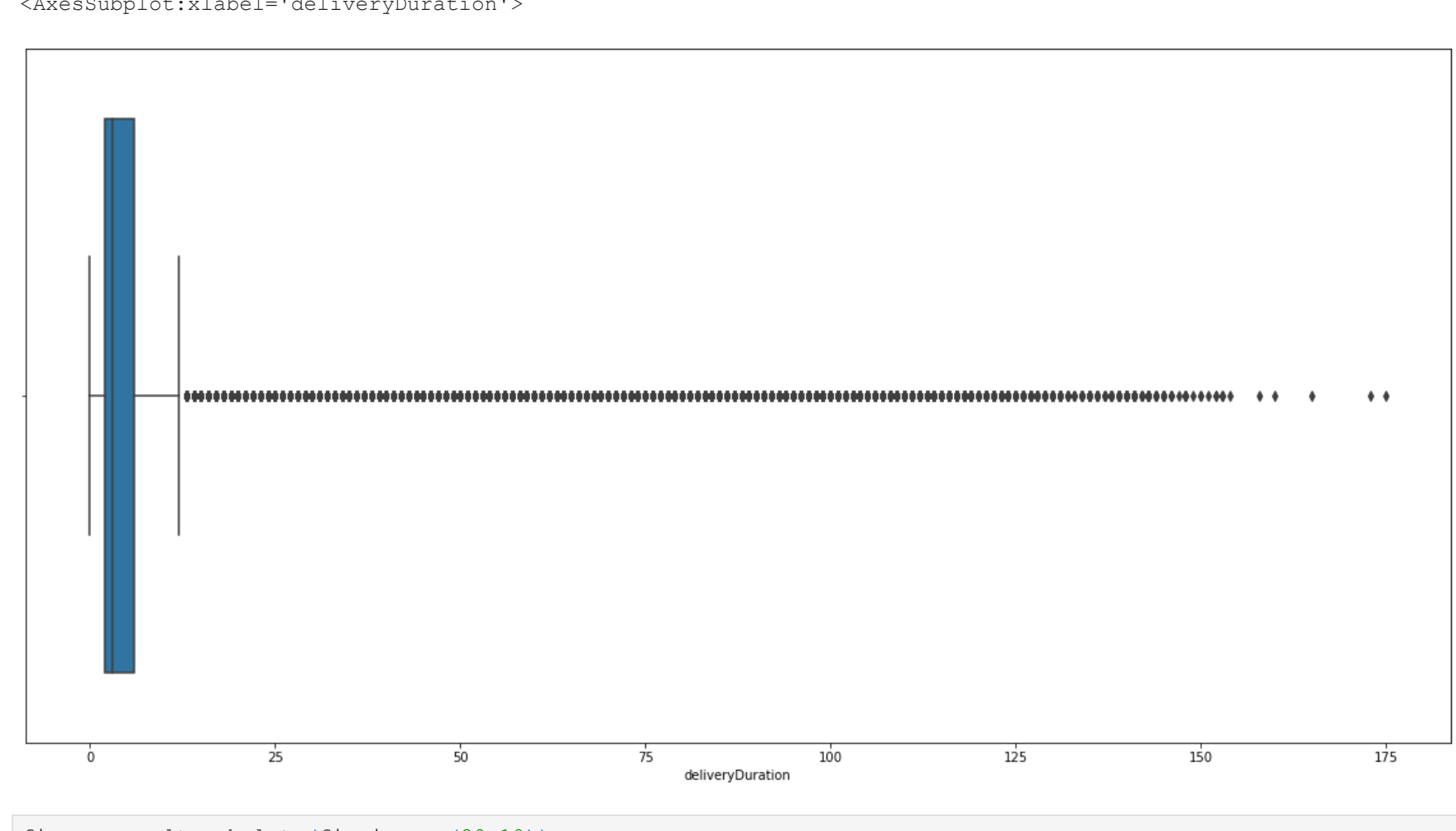
In [14]:
```
fig, ax = plt.subplots(figsize = (15,10))
sns.histplot(ax=ax, data=df2, x='deliveryDuration')
```

Out[14]: <AxesSubplot:xlabel='deliveryDuration', ylabel='Count'>



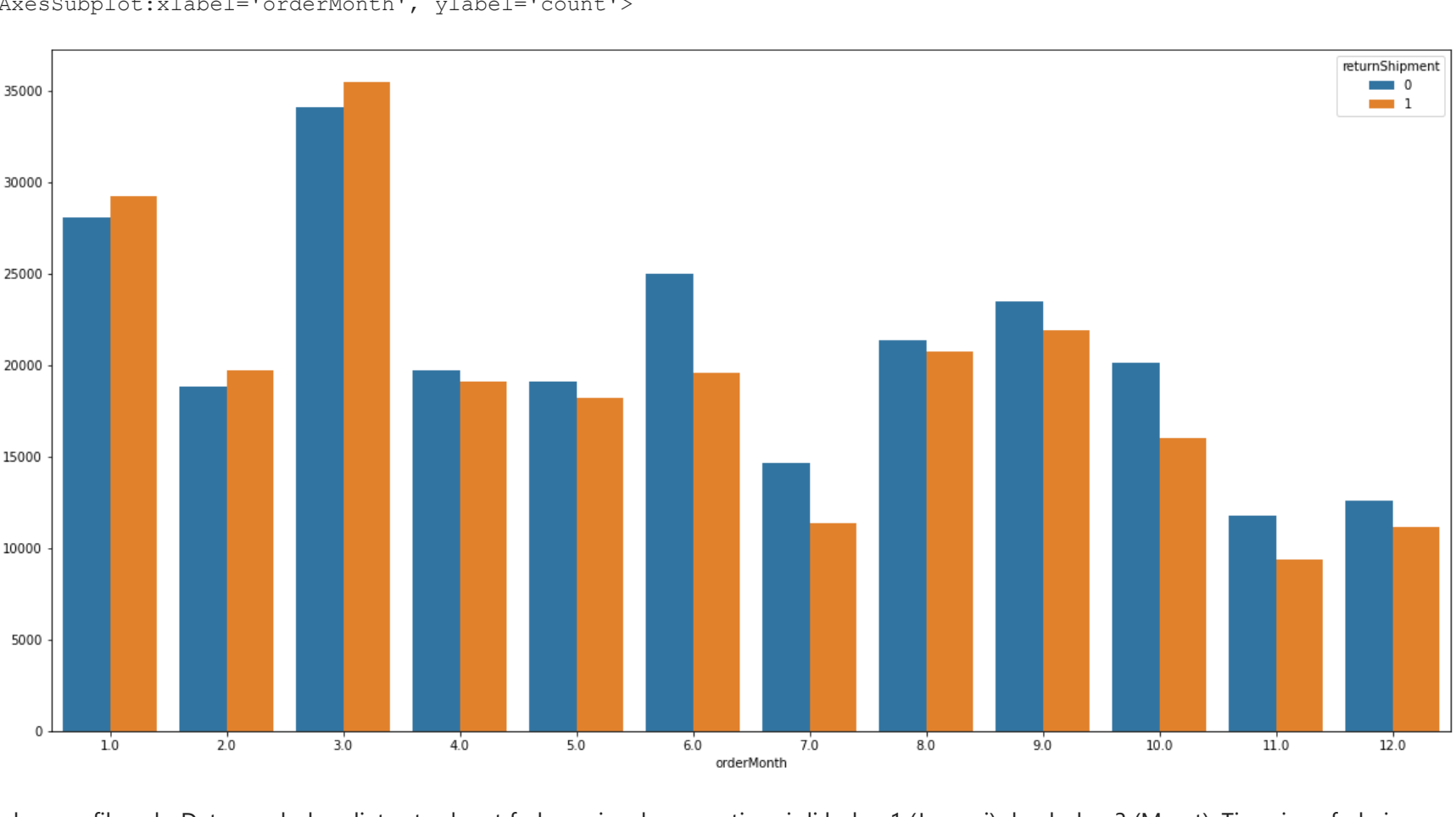
In [15]:
```
fig, ax = plt.subplots(figsize = (20,10))
sns.boxplot(ax=ax, x="deliveryDuration", data=df2)
```

Out[15]: <AxesSubplot:xlabel='deliveryDuration'>



In [16]:
```
fig, ax = plt.subplots(figsize = (20,10))
sns.countplot(ax=ax, x=df2['orderMonth'], hue='returnShipment', data=df2)
```

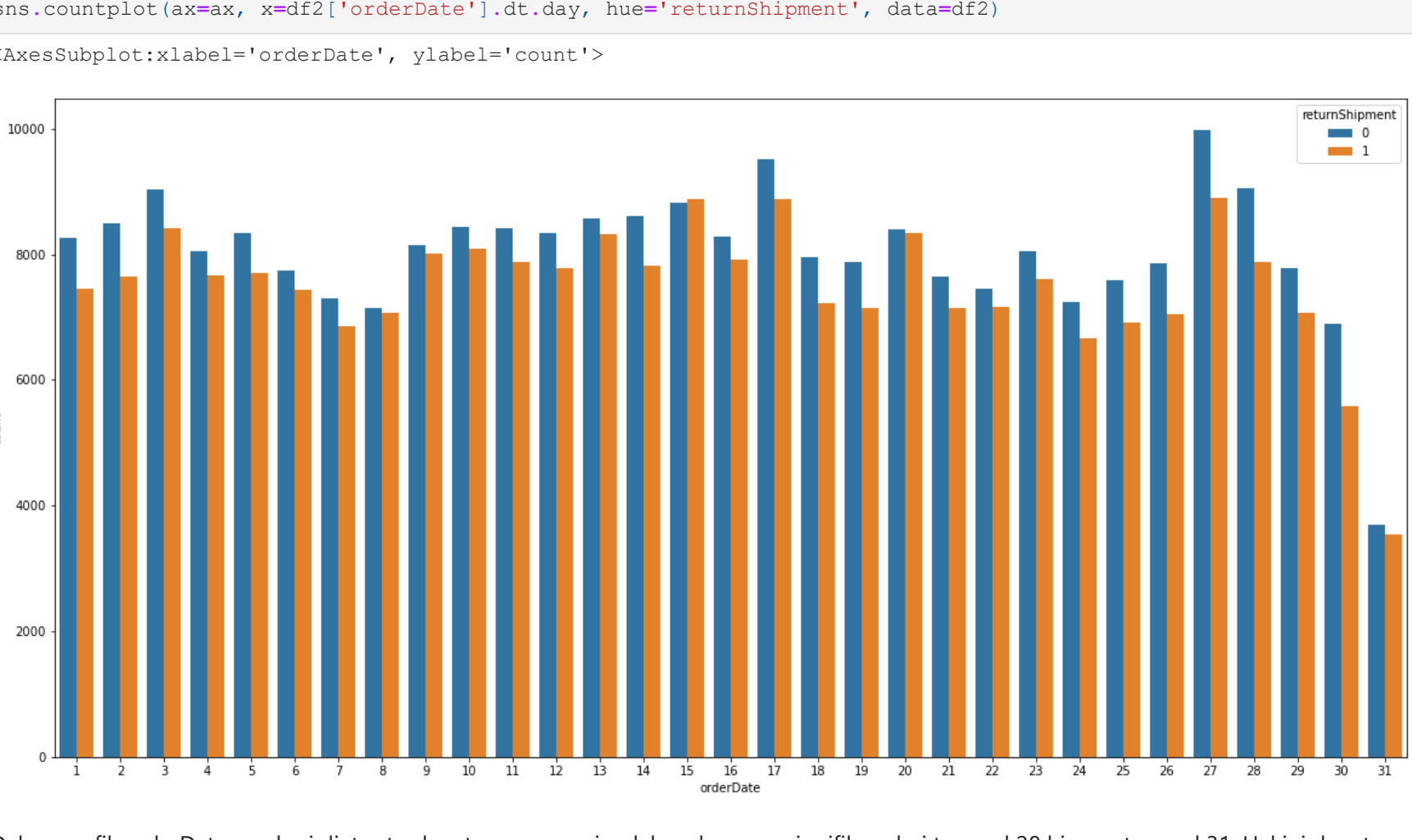
Out[16]: <AxesSubplot:xlabel='orderMonth', ylabel='count'>



Dalam grafik orderDate per-bulan diatas terdapat frekuensi order yang tinggi di bulan 1 (Januari) dan bulan 3 (Maret). Tingginya frekuins pada bulan 1 dapat menandakan bahwa setelah tahun baru terdapat peningkatan order yang drastis. Lalu ada peningkatan lagi di bulan 3 yang merupakan awal dari musim semi.

In [17]:
```
fig, ax = plt.subplots(figsize = (20,10))
sns.countplot(ax=ax, x=df2['orderDate'].dt.day, hue='returnShipment', data=df2)
```

Out[17]: <AxesSubplot:xlabel='orderDate', ylabel='count'>



Dalam grafik orderDate per-hari diatas terdapat penurunan jumlah order yang signifikan dari tanggal 28 hingga tanggal 31. Hal ini dapat disebabkan oleh jumlah tanggal pada bulan Februari yang hanya sampai 28 dan tidak semua bulan memiliki 31 hari.

Terdapat satu instance dimana jumlah pengembalian lebih banyak dari yang tidak mengembalikan di tanggal 15.

In [18]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 481092 entries, 0 to 481091
Data columns (total 21 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   orderItemID       481092 non-null  int64
 1   orderDate         481092 non-null  datetime64[ns]
 2   orderYear         481092 non-null  float64
 3   orderMonth        481092 non-null  float64
 4   orderDated        481092 non-null  float64
 5   deliveryDate      481092 non-null  datetime64[ns]
 6   deliveryYear      481092 non-null  float64
 7   deliveryMonth     481092 non-null  float64
 8   deliveryDated     481092 non-null  float64
 9   deliveryDuration  481092 non-null  float64
 10  itemID            481092 non-null  int64
 11  size              481092 non-null  object
 12  color             480949 non-null  object
 13  manufacturerID    481092 non-null  int64
 14  price             481092 non-null  float64
 15  customerID        481092 non-null  int64
 16  salutation        481092 non-null  object
 17  dateOfBirth       432200 non-null  datetime64[ns]
 18  state             481092 non-null  object
 19  creationDate      481092 non-null  datetime64[ns]
 20  returnShipment    481092 non-null  int64
dtypes: datetime64[ns](4), float64(8), int64(5), object(4)
memory usage: 77.1+ MB
```

praproses size

In [19]:
```
#mengganti data size yang hurufnya kecil menjadi huruf kapital agar seragam
df2.loc[(df2["size"] == "s"), "size"] = "S"
df2.loc[(df2["size"] == "m"), "size"] = "M"
df2.loc[(df2["size"] == "l"), "size"] = "L"
df2.loc[(df2["size"] == "xl"), "size"] = "XL"
df2.loc[(df2["size"] == "xs"), "size"] = "XS"
df2.loc[(df2["size"] == "xxl"), "size"] = "XXL"
df2.loc[(df2["size"] == "xxxl"), "size"] = "XXXL"
```

In [20]:
```
#mengganti data size yang belakangnya terdapat tanda '+' dengan huruf untuk mengurangi variasi data
df2.loc[(df2["size"] == "9+"), "size"] = "M"
df2.loc[(df2["size"] == "10+"), "size"] = "L"
df2.loc[(df2["size"] == "8+"), "size"] = "M"
df2.loc[(df2["size"] == "42+"), "size"] = "L"
df2.loc[(df2["size"] == "4+"), "size"] = "XS"
df2.loc[(df2["size"] == "6+"), "size"] = "S"
df2.loc[(df2["size"] == "7+"), "size"] = "S"
df2.loc[(df2["size"] == "36+"), "size"] = "XS"
df2.loc[(df2["size"] == "39+"), "size"] = "S"
df2.loc[(df2["size"] == "40+"), "size"] = "M"
df2.loc[(df2["size"] == "5+"), "size"] = "XS"
df2.loc[(df2["size"] == "38+"), "size"] = "S"
df2.loc[(df2["size"] == "11+"), "size"] = "L"
df2.loc[(df2["size"] == "37+"), "size"] = "XS"
df2.loc[(df2["size"] == "41+"), "size"] = "M"
df2.loc[(df2["size"] == "3+"), "size"] = "XXS"
df2.loc[(df2["size"] == "44+"), "size"] = "XL"
df2.loc[(df2["size"] == "43+"), "size"] = "L"
df2.loc[(df2["size"] == "13+"), "size"] = "XL"
df2.loc[(df2["size"] == "45+"), "size"] = "XL"
df2.loc[(df2["size"] == "46+"), "size"] = "XXL"
df2.loc[(df2["size"] == "2+"), "size"] = "XXS"
df2.loc[(df2["size"] == "12+"), "size"] = "XL"
```

In [21]:
```
#mengganti huruf menjadi angka pada atribut size untuk mengubah tipe datanya
df2.loc[(df2["size"] == "S"), "size"] = "38"
df2.loc[(df2["size"] == "M"), "size"] = "40"
df2.loc[(df2["size"] == "L"), "size"] = "42"
df2.loc[(df2["size"] == "XL"), "size"] = "44"
df2.loc[(df2["size"] == "XS"), "size"] = "36"
df2.loc[(df2["size"] == "XXS"), "size"] = "34"
df2.loc[(df2["size"] == "XXL"), "size"] = "46"
df2.loc[(df2["size"] == "XXXL"), "size"] = "48"
df2.loc[(df2["size"] == "unsized"), "size"] = "0"
```

In [22]:
```
#mengganti tipe data size menjadi integer
df2["size"] = df2["size"].astype("int64")
#membuat kolom baru bertipe object untuk menampung data hasil perubahan dari atribut size
df2.insert(5,"realsize", "-")
```

In [23]:
```
#mengganti angka menjadi huruf berdasarkan ukuran baju US
df2.loc[(df2["size"] >= 1) & (df2["size"] <= 3), "realsize"] = "XXS"
df2.loc[(df2["size"] >= 4) & (df2["size"] <= 5), "realsize"] = "XS"
df2.loc[(df2["size"] >= 6) & (df2["size"] <= 7), "realsize"] = "S"
df2.loc[(df2["size"] >= 8) & (df2["size"] <= 9), "realsize"] = "M"
df2.loc[(df2["size"] >= 10) & (df2["size"] <= 11), "realsize"] = "L"
df2.loc[(df2["size"] >= 12) & (df2["size"] <= 13), "realsize"] = "XL"
df2.loc[(df2["size"] >= 14) & (df2["size"] <= 15), "realsize"] = "XXL"
df2.loc[(df2["size"] >= 16) & (df2["size"] <= 17), "realsize"] = "XXXL"
```

In [24]:
```
#mengganti angka menjadi huruf berdasarkan ukuran baju Eropa
df2.loc[(df2["size"] >= 18) & (df2["size"] <= 35), "realsize"] = "XXS"
df2.loc[(df2["size"] >= 36) & (df2["size"] <= 37), "realsize"] = "XS"
df2.loc[(df2["size"] >= 38) & (df2["size"] <= 39), "realsize"] = "S"
df2.loc[(df2["size"] >= 40) & (df2["size"] <= 41), "realsize"] = "M"
df2.loc[(df2["size"] >= 42) & (df2["size"] <= 43), "realsize"] = "L"
df2.loc[(df2["size"] >= 44) & (df2["size"] <= 45), "realsize"] = "XL"
df2.loc[(df2["size"] >= 46) & (df2["size"] <= 47), "realsize"] = "XXL"
df2.loc[(df2["size"] >= 48), "realsize"] = "XXXL"
```

In [25]:
```
#mengubah data pada size yang 0 menjadi unsized
df2.loc[(df2["size"] == 0), "realsize"] = "unsized"
```

In [26]:
```
#menghapus kolom size
df2 = df2.drop(['size'],axis=1)
```

In [27]:
```
#mengganti nama kolom realsize menjadi size
df2.rename(columns = {'realsize':'size'}, inplace = True)
```

In [28]:
```
df2["size"].value_counts(dropna = False)
```

Out[28]:
```
M          116780
L           96052
S           84923
XL          69884
XXL         52434
unsized     26039
XS          19190
XXS          8514
XXXL         7276
Name: size, dtype: int64
```

In [29]:
```
df2.isna().sum()
```

Out[29]:
```
orderItemID             0
orderDate               0
orderYear               0
orderMonth              0
orderDated              0
size                    0
deliveryDate            0
deliveryYear            0
deliveryMonth           0
deliveryDated           0
deliveryDuration        0
itemID                  0
color                 143
manufacturerID          0
price                   0
customerID              0
salutation              0
dateOfBirth         48892
state                   0
creationDate            0
returnShipment          0
dtype: int64
```

praproses color

In [30]:
```
#Proses Smoothing warna menjadi hanya warna primer, sekunder, dan netral
df2["color"] = df2["color"].replace('basalt', 'black')
df2["color"] = df2["color"].replace('ebony', 'black')
df2["color"] = df2["color"].replace('aqua', 'blue')
df2["color"] = df2["color"].replace('azure', 'blue')
df2["color"] = df2["color"].replace('blau', 'blue')
df2["color"] = df2["color"].replace('baltic blue', 'blue')
df2["color"] = df2["color"].replace('cobalt blue', 'blue')
df2["color"] = df2["color"].replace('dark denim', 'blue')
df2["color"] = df2["color"].replace('dark navy', 'blue')
df2["color"] = df2["color"].replace('darkblue', 'blue')
df2["color"] = df2["color"].replace('denim', 'blue')
df2["color"] = df2["color"].replace('navy', 'blue')
df2["color"] = df2["color"].replace('petrol', 'blue')
df2["color"] = df2["color"].replace('almond', 'brown')
df2["color"] = df2["color"].replace('bronze', 'brown')
df2["color"] = df2["color"].replace('brwon', 'brown')
df2["color"] = df2["color"].replace('caramel', 'brown')
df2["color"] = df2["color"].replace('cognac', 'brown')
df2["color"] = df2["color"].replace('copper coin', 'brown')
df2["color"] = df2["color"].replace('cortina mocca', 'brown')
df2["color"] = df2["color"].replace('khaki', 'brown')
df2["color"] = df2["color"].replace('mahagoni', 'brown')
df2["color"] = df2["color"].replace('mocca', 'brown')
df2["color"] = df2["color"].replace('ocher', 'brown')
df2["color"] = df2["color"].replace('terracotta', 'brown')
df2["color"] = df2["color"].replace('aquamarine', 'green')
df2["color"] = df2["color"].replace('avocado', 'green')
df2["color"] = df2["color"].replace('dark oliv', 'green')
df2["color"] = df2["color"].replace('jade', 'green')
df2["color"] = df2["color"].replace('mint', 'green')
df2["color"] = df2["color"].replace('oliv', 'green')
df2["color"] = df2["color"].replace('olive', 'green')
df2["color"] = df2["color"].replace('turquoise', 'green')
df2["color"] = df2["color"].replace('anthracite', 'grey')
df2["color"] = df2["color"].replace('ash', 'grey')
df2["color"] = df2["color"].replace('dark grey', 'grey')
df2["color"] = df2["color"].replace('graphite', 'grey')
df2["color"] = df2["color"].replace('iron', 'grey')
df2["color"] = df2["color"].replace('apricot', 'orange')
df2["color"] = df2["color"].replace('coral', 'orange')
df2["color"] = df2["color"].replace('mango', 'orange')
df2["color"] = df2["color"].replace('antique pink', 'pink')
df2["color"] = df2["color"].replace('fuchsia', 'pink')
df2["color"] = df2["color"].replace('magenta', 'pink')
df2["color"] = df2["color"].replace('amethyst', 'purple')
df2["color"] = df2["color"].replace('aubergine', 'purple')
df2["color"] = df2["color"].replace('berry', 'purple')
df2["color"] = df2["color"].replace('currant purple', 'purple')
df2["color"] = df2["color"].replace('hibiscus', 'purple')
df2["color"] = df2["color"].replace('bordeaux', 'red')
df2["color"] = df2["color"].replace('crimson', 'red')
df2["color"] = df2["color"].replace('dark garnet', 'red')
df2["color"] = df2["color"].replace('beige', 'white')
df2["color"] = df2["color"].replace('champagner', 'white')
df2["color"] = df2["color"].replace('ecru', 'white')
df2["color"] = df2["color"].replace('ivory', 'white')
df2["color"] = df2["color"].replace('silver', 'white')
df2["color"] = df2["color"].replace('vanille', 'white')
df2["color"] = df2["color"].replace('creme', 'yellow')
df2["color"] = df2["color"].replace('curry', 'yellow')
df2["color"] = df2["color"].replace('gold', 'yellow')
df2["color"] = df2["color"].replace('ingwer', 'yellow')
df2["color"] = df2["color"].replace('lemon', 'yellow')
#Proses smoothing untuk warna yang merupakan sebuah pola dan atau memiliki lebih dari 1 warna dasar
df2["color"] = df2["color"].replace('leopard', 'other')
df2["color"] = df2["color"].replace('perlmutt', 'other')
df2["color"] = df2["color"].replace('curled', 'other')
df2["color"] = df2["color"].replace('aviator', 'other')
df2["color"] = df2["color"].replace('opal', 'other')
df2["color"] = df2["color"].replace('kanel', 'other')
df2["color"] = df2["color"].replace('striped', 'other')
df2["color"] = df2["color"].replace('floral', 'other')
df2["color"] = df2["color"].replace('stained', 'other')
df2["color"] = df2["color"].replace('ancient', 'other')
df2["color"] = df2["color"].replace('pallid', 'other')
df2["color"] = df2["color"].replace('nature', 'other')
df2["color"] = df2["color"].replace('habana', 'other')
```

In [31]:
```
#Proses mengisi entry yang NaN menjadi "unspecified"
df2.loc[(df2["color"].isna()), "color"] = "unspecified"
```

In [32]:
```
df2.isna().sum()
```

Out[32]:
```
orderItemID             0
orderDate               0
orderYear               0
orderMonth              0
orderDated              0
size                    0
deliveryDate            0
deliveryYear            0
deliveryMonth           0
deliveryDated           0
deliveryDuration        0
itemID                  0
color                   0
manufacturerID          0
price                   0
customerID              0
salutation              0
dateOfBirth         48892
state                   0
creationDate            0
returnShipment          0
dtype: int64
```

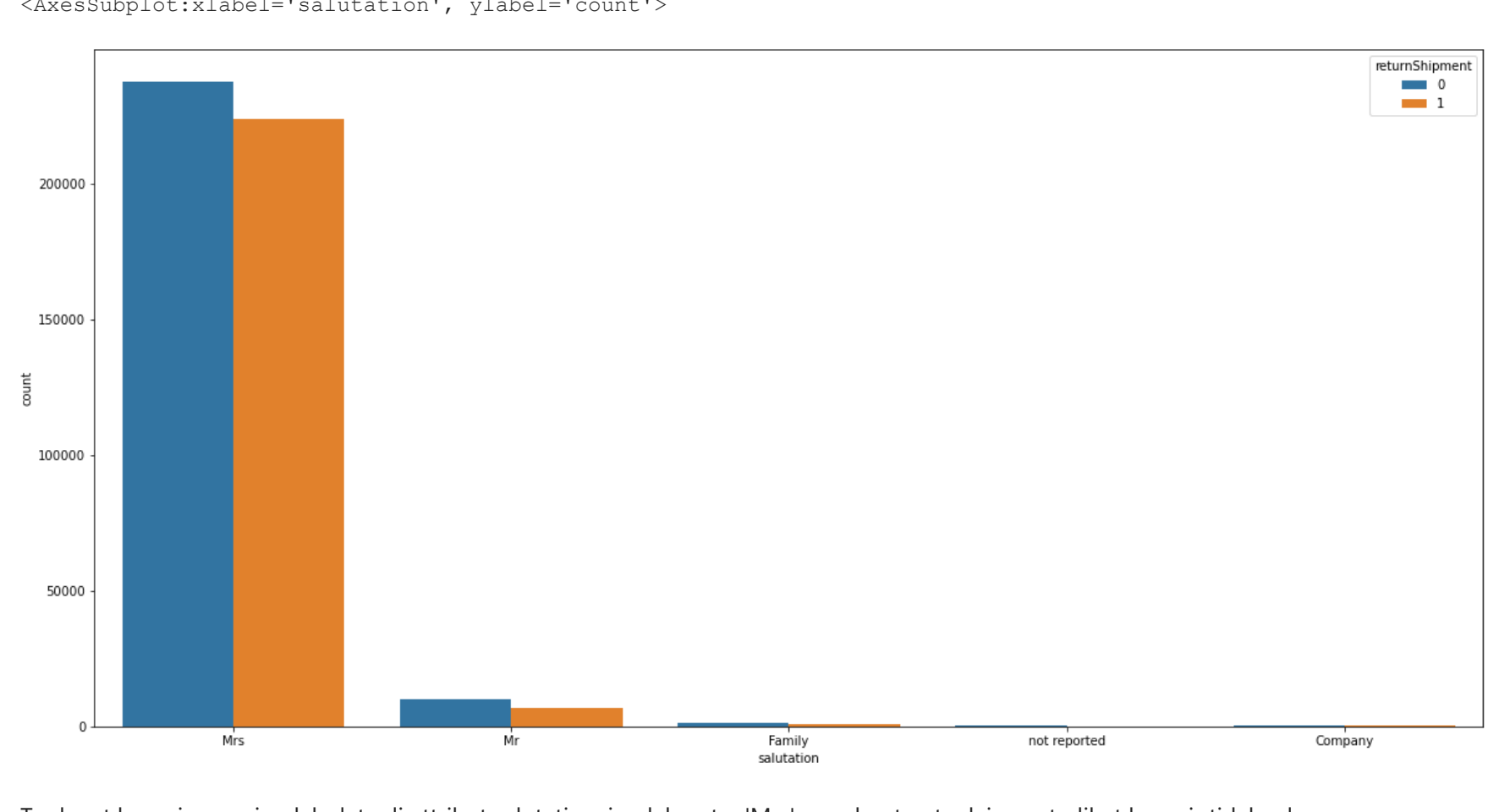
In [33]:
```
df2['salutation'].value_counts(dropna=False)
```

Out[33]:
```
Mrs             461769
Mr               16720
Family            1891
Company            361
not reported       351
Name: salutation, dtype: int64
```

praproses salutation

In [34]:
```
fig, ax = plt.subplots(figsize = (20,10))
sns.countplot(ax=ax, x="salutation", hue='returnShipment', data=df2)
```

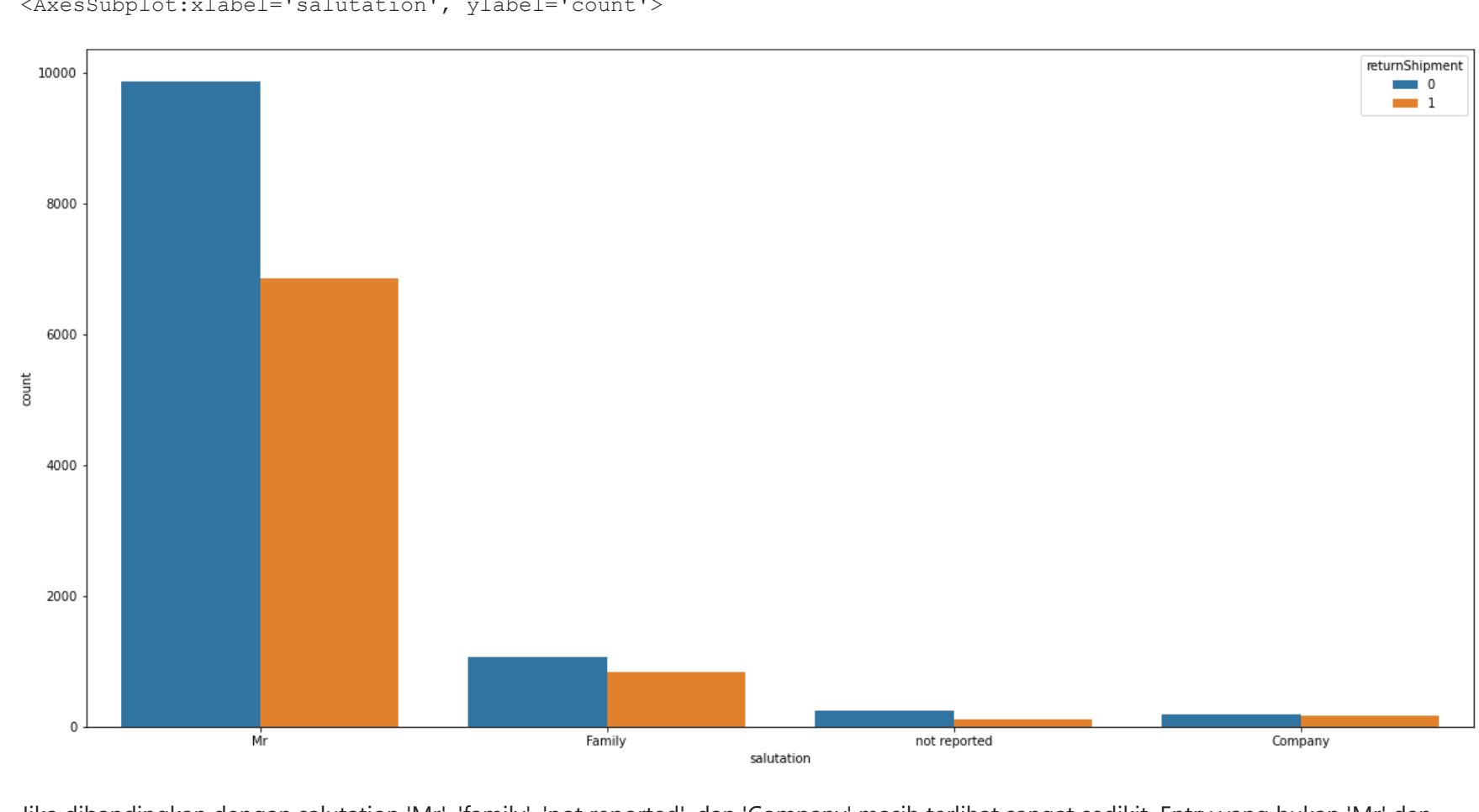
Out[34]: <AxesSubplot:xlabel='salutation', ylabel='count'>



Terdapat kesenjangan jumlah data di attribut salutation, jumlah entry 'Mrs' membuat entry lainnya terlihat hampir tidak ada

In [35]:
```
fig, ax = plt.subplots(figsize = (20,10))
sns.countplot(ax=ax, x=df2["salutation"].loc[(df2["salutation"] != "Mrs")], hue='returnShipment', data=df2)
```

Out[35]: <AxesSubplot:xlabel='salutation', ylabel='count'>



Jika dibandingkan dengan salutation 'Mr', 'family', 'not reported', dan 'Company' masih terlihat sangat sedikit. Entry yang bukan 'Mr' dan 'Mrs' dikelompokkan menjadi 'Other' karna jumlahnya yang sedikit dan agar lebih terfokus ke gendered 'salutation'.

In [36]:
```
df2['salutation'] = df2['salutation'].replace('Family', 'Other')
df2['salutation'] = df2['salutation'].replace('Company', 'Other')
df2['salutation'] = df2['salutation'].replace('not reported', 'Other')
```

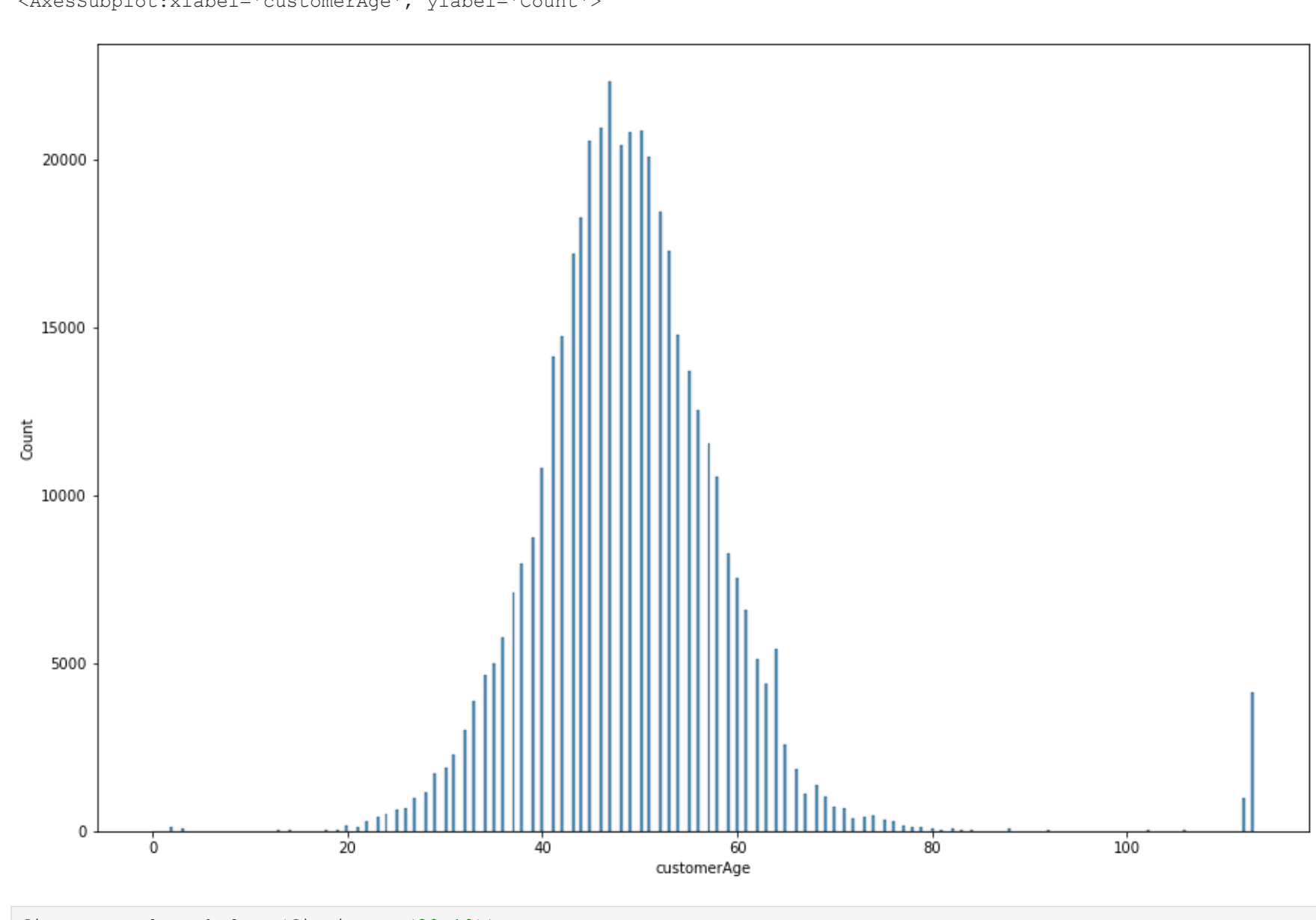
praproses dateOfBirth

In [37]:
```
df2.insert(19, "customerAge", np.nan)
```

In [38]:
```
df2["customerAge"] = df2["deliveryYear"].max() - df2["dateOfBirth"].dt.year
```

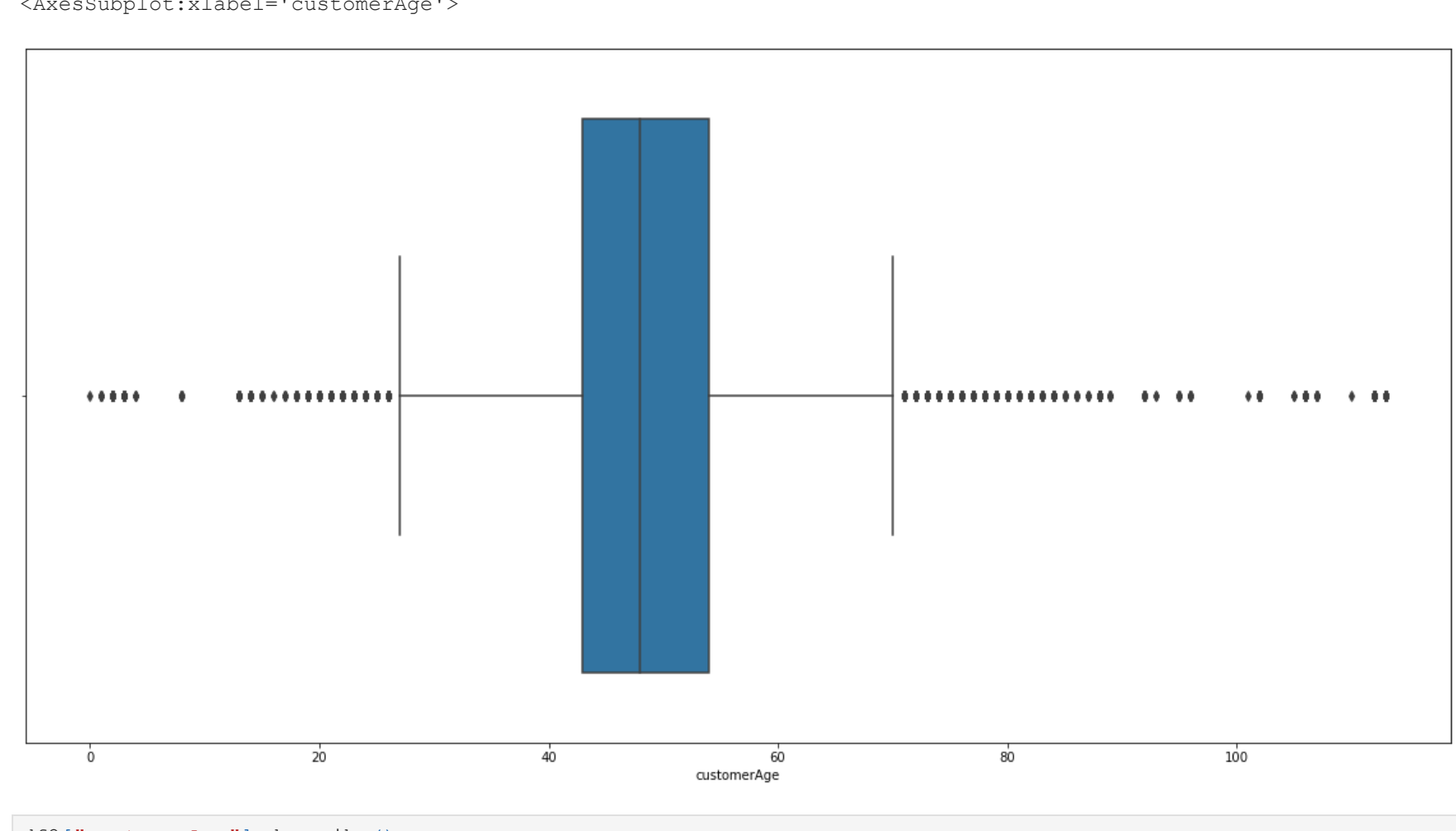
In [39]:
```
fig, ax = plt.subplots(figsize = (15,10))
sns.histplot(ax=ax, data=df2, x='customerAge')
```

Out[39]: <AxesSubplot:xlabel='customerAge', ylabel='Count'>



In [40]:
```
fig, ax = plt.subplots(figsize = (20,10))
sns.boxplot(ax=ax, x="customerAge", data=df2)
```

Out[40]: <AxesSubplot:xlabel='customerAge'>



In [41]:
```
df2["customerAge"].describe()
```

Out[41]:
```
count    432200.000000
mean         49.218061
std          11.065563
min           0.000000
25%          43.000000
50%          48.000000
75%          54.000000
max         113.000000
Name: customerAge, dtype: float64
```

In [42]:
```
df2["customerAge"].isna().sum()
```

Out[42]: 48892

Mengisi data customerAge yang kosong menggunakan median

In [43]:
```
df2.loc[(df2["customerAge"].isna()), "customerAge"] = df2["customerAge"].median()
```

deteksi outlier

IQR = 53 - 44

Batas atas = 53 + 1.5(11)

Batas bawah = 44 - 1.5(11)

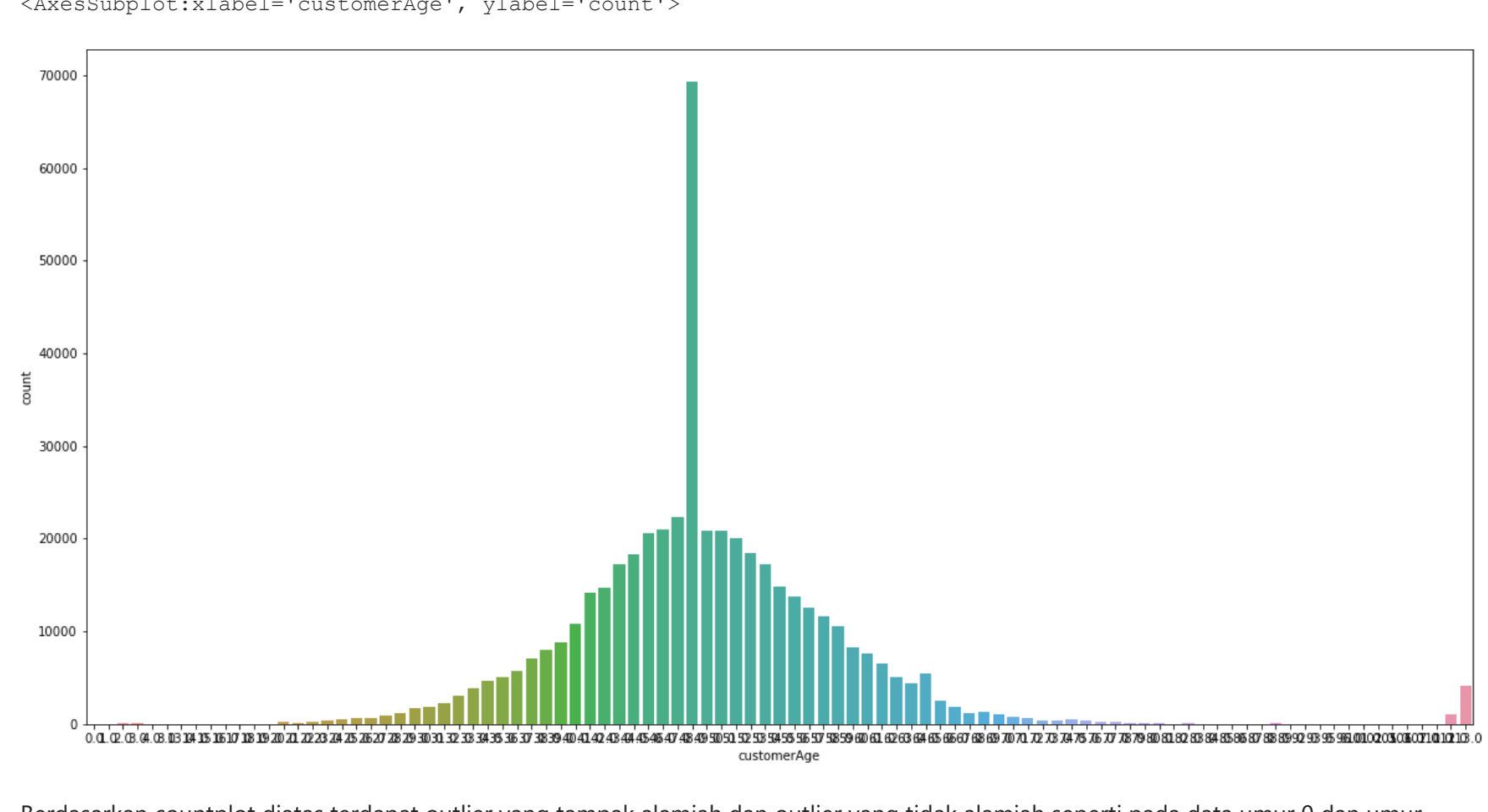
Batas atas = 69.5

Batas bawah = 27.5

In [44]:
```
upper_bound = 69
lower_bound = 27
```

In [45]:
```
fig, ax = plt.subplots(figsize = (20,10))
sns.countplot(ax=ax, x="customerAge", data=df2)
```

Out[45]: <AxesSubplot:xlabel='customerAge', ylabel='count'>



Berdasarkan countplot diatas terdapat outlier yang tampak alamiah dan outlier yang tidak alamiah seperti pada data umur 0 dan umur 112. Untuk menangani outlier tidak alamiah, yang berada di bawah batas umur penggunaan akun google (13 tahun) akan diubah menjadi nilai lower_bound, dan yang berada di atas angka harapan hidup negara Jerman di tahun 2013 (80 tahun) akan diubah menjadi nilai upper_bound.

In [46]:
```
#proses penanganan outlier diatas 80 tahun
df2.loc[((df2.customerAge > 80)), "customerAge"] = upper_bound

#proses penanganan outlier dibawah 13 tahun
df2.loc[((df2.customerAge < 13)), "customerAge"] = lower_bound
```

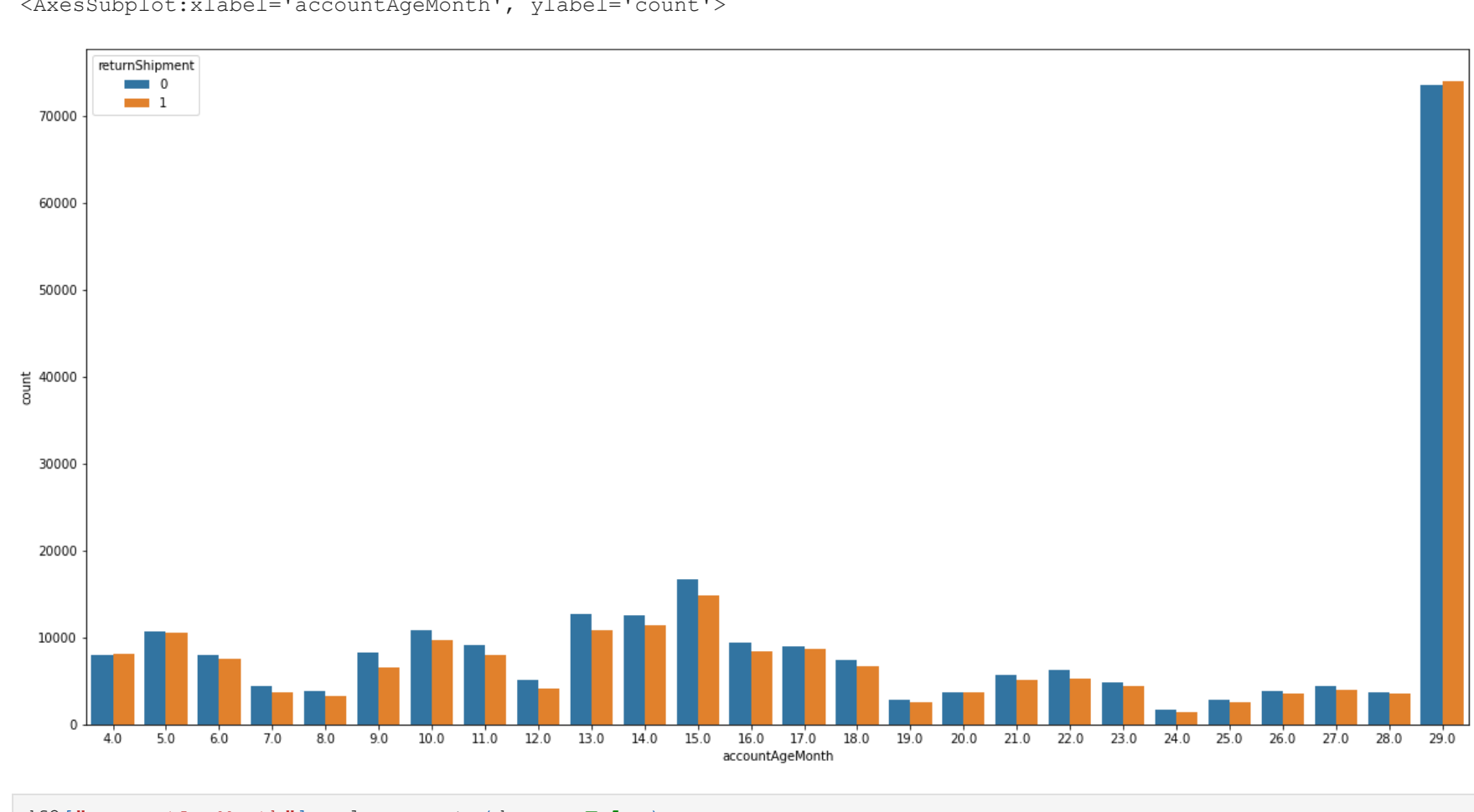
praproses creationDate

In [47]:
```
df2.insert(21, "accountAgeMonth", np.nan)
```

In [48]:
```
df2["accountAgeMonth"] = ((df2["deliveryDate"].max() - df2["creationDate"]) / np.timedelta64(1, 'M')).round()
```

In [49]:
```
fig, ax = plt.subplots(figsize = (20,10))
sns.countplot(ax=ax, x="accountAgeMonth", hue="returnShipment", data=df2)
```

Out[49]: <AxesSubplot:xlabel='accountAgeMonth', ylabel='count'>



In [50]:
```
df2["accountAgeMonth"].value_counts(dropna=False)
```

Out[100]:
```
array([1, 1, 1, ..., 0, 0, 1], dtype=int64)
```

In [101…
```
i = 1
with open("C:\\Users\\riska\\Downloads\\data_returns\\order_class.txt", "w") as o:
    print("{0};{1}".format("orderItemID", "prediction"), file = o)
    for line in clf.predict(df_class2):
        print("{0};{1}".format(i, line), file=o)
        i+=1
```