

CONTOH STUDY CASE

IMPLEMENTASI PENGGUNAAN MYSQL DAN MS. EXCEL DALAM PERENCANAAN STRATEGI BISNIS

Disusun Oleh:

Sefa Falahudin

DAFTAR ISI

DAFTAR ISI.....	ii
DAFTAR GAMBAR	iii
DAFTAR TABEL.....	iv
DAFTAR GRAFIK.....	v
A. RDBMS (<i>Relational Database Management System</i>).....	1
B. <i>Data Definition Language</i> (DDL)	2
C. <i>Data Manipulation Language</i> (DML)	2
1. Create Database.....	2
2. Create Table	2
3. Insert Into	3
4. Select * from	3
5. Desc.....	4
6. Alter table.....	4
D. Penerepan Penggunaan <i>Data Manipulation Language</i> (DML) Dalam Analisis Data	5
Case 1. Mencari total pendapatan dari pembelian setiap customer.....	5
Case 2. Total unit yang terjual per bulan.....	6
Case 3. Produk yang paling sering dibeli setiap customer	7
Case 4. Total Pendapatan Per Bulan.....	8
E. Strategi Pemasaran.....	10
Strategi 1.....	10
Strategi 2.....	12
DAFTAR PUSTAKA	14
LAMPIRAN.....	15

DAFTAR GAMBAR

Gambar 1. Diagram many to many relationships.....	1
Gambar 2. Command pembuatan table	3
Gambar 3. Fitur table data import	3
Gambar 4. Output penggunaan command select.....	4
Gambar 5. Output penggunaan command desc.....	4
Gambar 6. Contoh penggunaan command alter table	4
Gambar 7. Query untuk menjawab case 1	5
Gambar 8. Output hasil query pada case 1	6
Gambar 9. Query untuk menjawab case 2	6
Gambar 10. Output hasil query pada case 2.....	6
Gambar 11. Query untuk menjawab case 3	7
Gambar 12. Output hasil query pada case 3.....	7
Gambar 13. Query untuk menjawab case 4	8
Gambar 14. Output hasil query pada case 4.....	8
Gambar 15. Query untuk menjawab case 5	9
Gambar 16. Output hasil query pada case 4.....	9
Gambar 17. Hasil pengaplikasian what if analyst goal seek.....	12

DAFTAR TABEL

Tabel 1. Penawaran setiap tingkatan member.....	11
Tabel 2. Total pendapatan, unit terjual dan laba 15% setiap bulannya.....	12

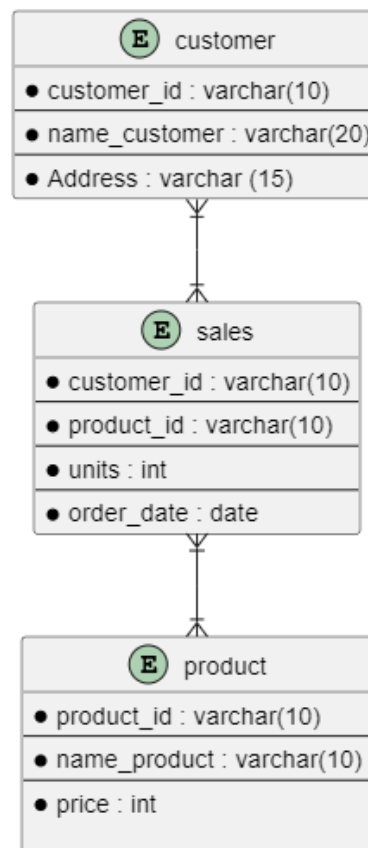
DAFTAR GRAFIK

Grafik 1. Visualisasi pencapaian target.....	10
Grafik 2. Visualisasi total pembelian setiap customer	10

A. RDBMS (*Relational Database Management System*)

Sebuah database memiliki cakupan informasi yang sangat besar yang tersimpan dalam bentuk dokumen atau tabel. DBMS membantu mendefinisikan di mana sebuah informasi terletak serta kaitannya dengan informasi di tabel atau dokumen lain. Relation pada gambar dibawah merupakan Many to Many Relationships. Many to Many adalah relasi dimana ada relasi antara dua tabel dimana table pertama bisa punya banyak relasi di table kedua, dan table kedua pun punya banyak relasi di table pertama.

Contoh relasi many to many adalah relasi antara produk dan penjualan, dimana setiap produk bisa dijual berkali kali, dan setiap penjualan bisa untuk lebih dari satu produk. Gambar 1 merupakan relations dibuat menggunakan **plantuml** yang merupakan table data base yang digunakan dan merupakan sebuah gambaran relasi yang akan kita gunakan dalam perintah join.



Gambar 1. Diagram many to many relationships

B. *Data Definition Language (DDL)*

Data Definition Language (DDL) merupakan Bahasa yang digunakan untuk membuat dan mengubah struktur dari sebuah obyek dalam database seperti views, schema, tabel, index, dsb.

C. *Data Manipulation Language (DML)*

Data Manipulation Language (DML): Bahasa ini digunakan untuk memberikan perintah untuk memanipulasi sebuah data di dalam database. Perintah tersebut meliputi pengambilan data, menambahkan data, mengubah data, dan menghapus data. Contoh basic command pada mysql sebagai berikut:

1. Create Database

Database adalah tempat menyimpan table di MySQL. Jika misalkan sebuah table di MySQL adalah sebuah file, maka database merupakan folder penyimpanannya, dimana kita bisa menyimpan banyak table di sebuah database. Command dalam pembuatan database adalah **create database nama_database ;**.

2. Create Table

Tiap tabel biasanya menyimpan satu jenis data, misal membuat Laporan penjualan, kita akan membuat tabel produk, tabel customer, tabel penjual, dan lain-lain. Sebelum memasukkan data ke tabel, wajib terlebih dahulu membuat tabelnya terlebih dahulu dan tiap tabel yang kita buat, wajib ditentukan kolom-kolom nya, dan tipe data tiap kolom nya. Selain membuat table di mysql juga bisa mengubah tabel yang sudah terlanjur dibuat, seperti menambah kolom baru, mengubah kolom yang sudah ada, atau menghapus kolom. Pada gambar 2 merupakan gambar command dalam pembuatan table.

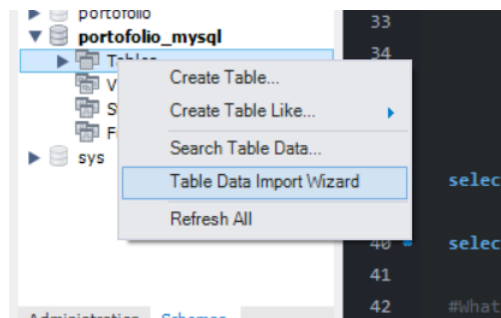
```

5
6 ■ create table product
7 (
8     product_id varchar(10),
9     name_product varchar(10),
10    price int unsigned
11 ) engine = innodb;
12
13 ■ insert into product (product_id, name_product, price)
14     values
15         ('G32N', 'gaun', 400000),
16         ('J22S', 'jas', 350000),
17         ('K78A', 'kemeja', 250000),
18         ('M90L', 'mantel', 170000),
19         ('R28K', 'rok', 85000);
20 ■ select * from product;
21

```

Gambar 2. Command pembuatan table

Selain menggunakan command create table dalam pembuatan table, kita juga dapat mengimport tabel jika tabel tersebut sudah dibuat menggunakan excel. Yaitu dengan menggunakan fitur **Table Data Import Wizard** yang dapat dilihat pada gambar 3. Perlu diingat file yang di import harus dalam format **csv**.



Gambar 3. Fitur table data import

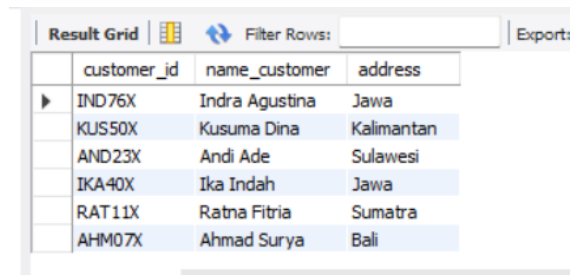
3. Insert Into

Pada **gambar 3 baris ke 13** terlihat terdapat command **insert into**. Command tersebut berfungsi untuk memasukan header atau kolom pada tabel. Kemudian command **Values** berfungsi untuk memasukan data pada setiap kolom dan penulisan datanya harus diurutkan sesuai nama kolom.

4. Select * from

Select from sendiri merupakan command yang digunakan untuk menampilkan hasil dari pembuatan tabel. Dapat dilihat pada gambar 2 baris ke-20. Jika ingin menampilkan beberapa kolom yang akan dipilih dapat ditulis

nama kolom tersebut tanpa menggunakan tanda bintang (*). Gambar 4 merupakan contoh output penggunaan command **select from**.



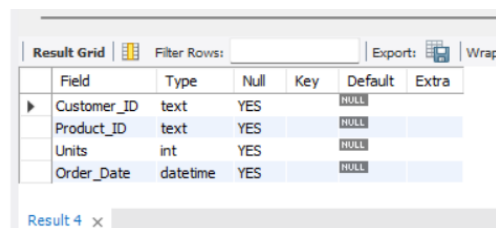
The screenshot shows a 'Result Grid' window with a table containing customer information. The table has three columns: 'customer_id', 'name_customer', and 'address'. There are six rows of data.

customer_id	name_customer	address
IND76X	Indra Agustina	Jawa
KUS50X	Kusuma Dina	Kalimantan
AND23X	Andi Ade	Sulawesi
IKA40X	Ika Indah	Jawa
RAT11X	Ratna Fitria	Sumatra
AHM07X	Ahmad Surya	Bali

Gambar 4. Output penggunaan command select

5. Desc

Jika ingin melihat format suatu tabel kita dapat menggunakan command **desc nama_column**. Tentunya command ini berbeda dengan select, yang mana select menampilkan isi table sedangkan desc menampilkan format setiap kolom pada tabel. Gambar 5 merupakan contoh dari output penggunaan command desc.



The screenshot shows a 'Result Grid' window displaying the structure of a table. The table has six columns: 'Field', 'Type', 'Null', 'Key', 'Default', and 'Extra'. There are four rows of data representing different fields in the table.

Field	Type	Null	Key	Default	Extra
Customer_ID	text	YES		NULL	
Product_ID	text	YES		NULL	
Units	int	YES		NULL	
Order_Date	datetime	YES		NULL	

Gambar 5. Output penggunaan command desc

6. Alter table

Alter tables merupakan sebuah fungsi untuk memodifikasi suatu nama kolom ataupun value dari kolom tersebut. Selain itu alter table juga dapat digunakan untuk memodifikasi struktur kolom, missal akan mengubah bentuk varchar menjadi int.

```
ALTER TABLE barang
ADD COLUMN nama_column TEXT,
DROP COLUMN nama,
RENAME COLUMN nama TO nama_baru,
MODIFY nama VARCHAR(100) AFTER jumlah,
MODIFY nama VARCHAR(100) FIRST;
```

Gambar 6. Contoh penggunaan command alter table

D. Penerepan Penggunaan *Data Manipulation Language* (DML) Dalam Analisis Data

Data yang digunakan pada contoh penerapan analisi data berisi table pada gambar 1 yaitu table sales, customer dan product. Contoh case dalam penerapan analisis data sebagai berikut:

Case 1. Mencari total pendapatan dari pembelian setiap customer

Supaya dapat menjawab pertanyaan tersebut dapat menggunakan CTE (*Common Table Expression*), CTE berfungsi untuk mendapatkan total biaya hal tersebut dikarenakan dapat dilihat pada gambar 1 bahwa colom unit dan price terletak pada tabel yang berbeda. Sehingga perlu menggunakan CTE yang dikombinasikan dengan join supaya mendapatkan hasil total biaya.

Permasalahan lainnya adalah hasil dari CTE tersebut bukan hasil setiap customer, maka dari itu kita perlu menjumlahkan hasil dari CTE yang berdasarkan nama customer. Supaya penjumlahan tersebut dapat based on customer maka perlu menggunakan command group by. Selain itu supaya outputnya mengurutkan berdasarkan total biaya tertinggi dapat menggunakan command order by nama table desc. Gambar 7 merupakan query yang digunakan supaya dapat mencari total biaya yang dikeluarkan setiap customer

```
45
46 with total_price as (
47     select name_customer, units*price as total from sales
48     join customer using(customer_id)
49     join product using(product_id))
50 select name_customer, sum(total) as total_amount
51 from customer
52 join total_price using(name_customer)
53 group by name_customer
54 order by total_amount desc;
55
```

Gambar 7. Query untuk menjawab case 1

Gambar 8 merupakan output dari query diatas. Dari gambar tersebut dapat diketahui bahwa total pendapatan paling banyak dari customer atas nama **Andi Ade** dengan total pendapatan **Rp. 261,530,000**.

Result Grid  Filter Rows: <input type="text"/>		
	name_customer	total_amount
▶	Andi Ade	Rp261530000
	Ika Indah	Rp248735000
	Ratna Fitria	Rp207800000
	Indra Agustina	Rp201815000
	Ahmad Surya	Rp201440000
	Kusuma Dina	Rp164760000

Gambar 8. Output hasil query pada case 1

Case 2. Total unit yang terjual per bulan

Untuk mencari total unit yang terjual query yang dibuat tidak sekompleks query untuk mencari total biaya yang dikeluarkan setiap customer. Hal tersebut dikarekan kita tidak perlu mengalikan atau menjumlah kolom dengan tabel yang berbeda. Kita cukup menjumlahkan unit berdasarkan date dan kedua kolom tersebut pada tabel yang sama yaitu tabel sales, maka dari itu kita tidak perlu menggunakan command join.

```

126 * select month(order_date) as month_order,
127      sum(units) as total_unit
128      from sales
129      group by month_order
130      order by month_order asc;

```

Gambar 9. Query untuk menjawab case 2

Berdasarkan gambar 10 dapat diketahui bahwa produk yang paling banyak terjual terdapat pada bulan 3 (Maret) unit 1,100 diikuti bulan 1 (Januari) dengan total unit 1,055. Sedangkan penjualan paling sedikit terdapat pada bulan 5 (Maret) dengan total penjualan 822 unit.

	month_order	total_unit
1	1	1055
2	2	979
3	3	1100
4	4	992
5	5	822

Gambar 10. Output hasil query pada case 2

Case 3. Produk yang paling sering dibeli setiap customer

Pada case 3 penggunaan CTE dikarenakan perlu mendapatkan hasil total produk berdasarkan pembelian customer. Penggunaan **common rank ()** bertujuan untuk memberi rank pada setiap jumlah total unit, jumlah unit yang paling banyak pada setiap customer akan berada pada rank 1. Kemudian supaya hanya rank 1 yang diambil perlu menggunakan command **where**.

```
64 with popular_item as
65 (
66     select name_customer, name_product, sum(units) as total_unit,
67           rank() over(partition by name_customer
68                     order by sum(units) desc) as rank_num
69     from sales
70     join customer using (customer_id)
71     join product using (product_id)
72     group by name_customer, name_product
73     order by name_customer asc, total_unit desc
74 )
75 select name_customer, name_product, total_unit
76 from popular_item
77 where rank_num = 1
78 order by total_unit desc;
```

Gambar 11. Query untuk menjawab case 3

Berdasarkan hasil query pada gambar 12, diketahui bahwa customer atas nama **Ika Indah** produk yang palings sering dibeli adalah **Jas**, **Andi Ade** dan **Kusuma Dina** adalah **Rok**, **Ahmad Surya** dan **Ratna Fitria** adalah **Gaun** dan **Indra Agustina** adalah **Mantel**.

	name_customer	name_product	total_unit
►	Ika Indah	jas	443
	Andi Ade	rok	402
	Ahmad Surya	gaun	301
	Ratna Fitria	gaun	284
	Indra Agustina	mantel	262
	Kusuma Dina	rok	212

Gambar 12. Output hasil query pada case 3

Case 4. Total Pendapatan Per Bulan

Untuk menjawab case 4, command yang digunakan hampir sama pada case 1. Yang pertama kita harus mengetahui pendapatan setiap penjualan product dengan command CTE, karena table sales dan price tiap product berbeda maka perlu melakukan join supaya dapat melakukan perkalian antara column units dan column price.

Selanjutnya supaya dapat mengetahui pendapatan perbulan kita dapat menjumlahkan hasil dari CTE yang kemudian group by berdasarkan month order. Month order sendiri didapatkan dengan cara mengekstrak date dengan menggunakan command **month(order_date)**, command tersebut terdapat pada gambar 13 baris ke-84.

```
82 • with total_price as
83 (
84     select month(order_date) as month_order, units*price as total
85     from sales
86     join product using (product_id)
87 )
88 select month_order, concat('Rp',sum(total)) as total_amount from total_price
89 group by month_order order by month_order asc, total_amount desc;
90
```

Gambar 13. Query untuk menjawab case 4

Pada gambar 14 dapat diketahui pendapatan per bulan yaitu, pada bulan 1 (Januari) memiliki pendapatan Rp. 271,060,000, bulan 2 (Februari) sampai bulan ke 4 (April) terus mengalami kenaikan dengan total pendapatan di bulan 2 (Februari) Rp. 271,405,000, bulan 3 (Maret) Rp. 272,150,000, bulan 4 (April) Rp. 284,900,000 dan pada bulan 5 (Mei) mengalami penurunan dengan total pendapatan Rp. 186,565,000.

	month_order	total_amount
▶	1	Rp271060000
	2	Rp271405000
	3	Rp272150000
	4	Rp284900000
	5	Rp186565000

Gambar 14. Output hasil query pada case 4

Case 5. Berapa Rata-Rata Pendapatan Per-Bulan

Sama halnya dengan case sebelumnya, CTE digunakan untuk mendapatkan total pendapatan. Yang membedakan adalah pada query gambar 15 baris ke 118 terdapat command avg yaitu untuk mendapatkan rata-rata yang kemudian di kombinasi dengan command round untuk mendapatkan hasil dalam bilangan bulat.

```
110 with total_price as
111 (
112     select month(order_date) as month_order,
113           units*price as total
114     from sales
115     join product using (product_id)
116 )
117 select month_order,
118        concat('Rp',round(avg(total),0))
119        as avg_income
120   from total_price
121   group by month_order
122   order by month_order
123         asc, avg_income desc;
```

Gambar 15. Query untuk menjawab case 5

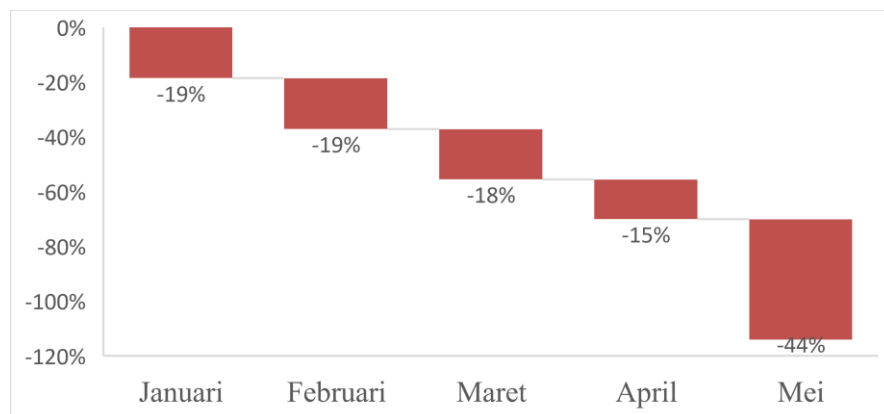
Dapat diketahui berdasarkan gambar 16, bahwa rata-rata pendapat 5 bulan terakhir cenderung terus mengalami fluktuatif. Yang mana nilai pendapatan pada 5 bulan terakhir sebagai berikut, bulan 1 (Januari) Rp. 8,743,871, bulan 2 (Februari) 9,693,036, bulan 3 (Maret) Rp. 8,779,032, bulan 4 (April) Rp. 9,496,667, bulan 5 (Mei) Rp. 8,111,522.

	month_order	avg_income
1	1	Rp8743871
2	2	Rp9693036
3	3	Rp8779032
4	4	Rp9496667
5	5	Rp8111522

Gambar 16. Output hasil query pada case 4

E. Strategi Pemasaran

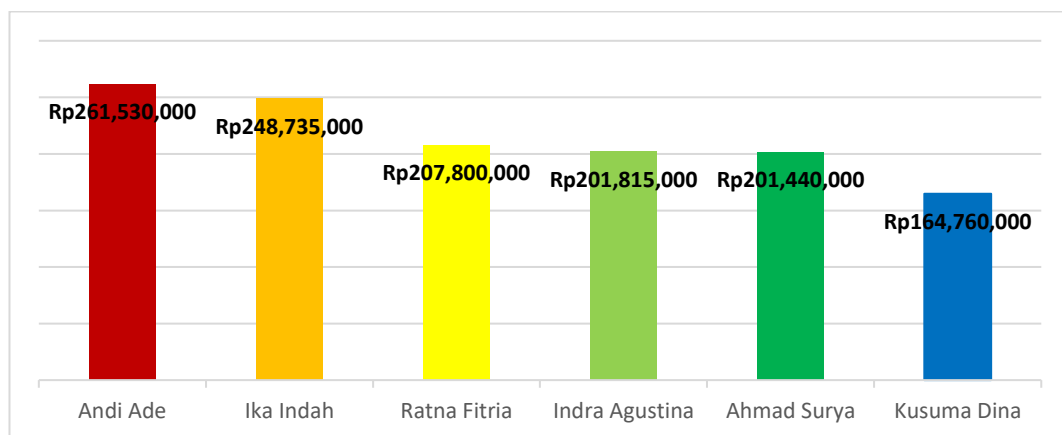
Setelah kita menganalisis data menggunakan Mysql, kita jauh akan lebih mudah untuk menentukan strategi pemasaran, supaya bisnis dapat berkembang. Namun hasil query diatas tentunya akan sulit untuk kita simpulkan, maka dari itu perlu adanya sebuah visualisasi data. Visualisasi data dibawah menggunakan data permisalan target perusahaan setiap bulannya adalah Rp 50,00,000, yang kemudian total laba 15% dari total pendapat perbulan dikurangi target dan dibagi target. Dari perhitungan tersebut didapatkan nilai persentase pencapaian.



Grafik 1. Visualisasi pencapaian target

Dari visualisasi data pada grafik 1 dapat diketahui bahwa dalam kurun waktu 5 bulan terakhir perusahaan tidak dapat mencapai target. Maka dari itu diperlukannya suatu strategi yang diharapkan pada bulan berikutnya dapat mencapai target yang diinginkan. Berikut merupakan strategi yang digunakan untuk dapat meningkatkan total pendapatan sehingga dapat mencapai target yang diinginkan.

Strategi 1



Grafik 2. Visualisasi total pembelian setiap customer

Berdasarkan data grafik 1. Kita bisa membuat sebuah strategi untuk mengembangkan pasar supaya dapat mempertahankan total pembelian pelanggan, bahkan menambah total pembelian setiap pelanggan. Salah satu strategi yang dapat digunakan dalam mengembangkan pemasaran adalah dengan konsep penjualan. Menurut Purwanto Eko (2020), konsep penjualan sendiri merupakan sebuah konsep yang meyakini bahwa kunci untuk mencapai tujuan perusahaan mencakup usaha perusahaan untuk menjadi lebih efektif dari para pesaing dalam hal menyampaikan kepada pasar sasaran yang ditetapkan nilai manfaat bagi konsumen.

Point yang bisa kita ambil dari statement diatas adalah memberi nilai manfaat bagi konsumen merupakan salah satu strategi yang efektif untuk mengembangkan target pemasaran. Salah satu strategi yang dapat digunakan dari grafik diatas adalah strategi status member, permisalan strateginya adalah member yang memiliki total pembelian >250jt merupakan member platinum, pembelian >200jt member gold dan pembelian <200jt member silver.

Tabel 1. Penawaran setiap tingkatan member

Member	Keuntungan
Platinum	Gratis ongkir dan diskon 10%
Gold	Gratis ongkir dan diskon 5%
Silver	Gratis ongkir

Tabel 1 merupakan sebuah penawaran berdasarkan tingkatan member, silver akan mendapatkan gratis ongkir, gold akan mendapat gratis ongkir dan diskon 5%, platinum akan mendapat gratis ongkir dan diskon 10%. Sistem pemberian status member sendiri dengan cara perhitungan total pemberian dengan range waktu 6 bulan, dan penawaran member tersebut akan berlaku untuk 6 bulan kedepan.

Strategi pemasaran diatas bertujuan supaya dapat memberi manfaat kepada konsumen, sehingga konsumen akan merasa diperhatikan atau dihargai. Selain itu diharapkan meningkatkan daya tarik konsumen, dimana konsumen belanja lebih banyak bahkan dapat memperluas pasar atau menambah konsumen.

Strategi 2

Tabel 2. Total pendapatan, unit terjual dan laba 15% setiap bulannya

Bulan	Total Pendapatan	Unit Terjual	Laba 15%
Januari	Rp271,060,000	1,055	Rp40,659,000
Februari	Rp271,405,000	979	Rp40,710,750
Maret	Rp272,150,000	1,100	Rp40,822,500
April	Rp284,900,000	992	Rp42,735,000
Mei	Rp186,565,000	822	Rp27,984,750
Target Laba			Rp50,000,000

Berdasarkan tabel 2 dapat diketahui bahwa dalam kurun waktu 5 bulan terakhir perusahaan tidak dapat mencapai target laba 15% yang diinginkan. Untuk dapat mencapai target laba perlu adanya peningkatan penjualan, supaya penjualan dapat meningkat kita dapat menggunakan strategi 1 yaitu dengan menarik pelanggan. Namun, yang menjadi permasalahannya berapakah unit yang harus terjual supaya dapat mencapai target laba, dan dalam produksi barang tidak kekurangan stok ataupun kelebihan stok sehingga dapat merugikan perusahaan.

Untuk mengatasi permasalahan diatas kita dapat menggunakan **what if analyst**, lebih tepatnya **goal seek** menggunakan Ms. Excel. Goal seek sendiri merupakan perintah untuk mencari berapa inputan atau nilai awal supaya mendapatkan output atau nilai akhir sesuai dengan target atau yang diharapkan.

Unit Terjual	Total Pendapatan	Laba 15%
1,055	Rp271,060,095	Rp40,659,014
979	Rp271,405,233	Rp40,710,785
1,100	Rp272,149,900	Rp40,822,485
992	Rp284,900,416	Rp42,735,062
822	Rp186,565,230	Rp27,984,785
target Laba		Rp50,000,000

Unit Te	Laba 15%
1,055	Rp40,659,014
979	Rp40,710,785
1,347	Rp50,000,000
992	Rp42,735,062
822	Rp27,984,785
target Laba	Rp50,000,000

Gambar 17. Hasil pengaplikasian what if analyst goal seek

Gambar 17 merupakan hasil dari analisis menggunakan goal seek, berdasarkan gambar tersebut dapat diketahui bahwa supaya dapat mencapai target

maka unit yang harus terjual harus sebanyak 1,347. Pada penggunaan analisis goal seek kenapa values yang digunakan adalah unit terjual yang paling banyak, hal tersebut dikarenakan seperti yang bisa kita lihat bahwa walaupun unit terjual paling tinggi namun laba dari total pendapatan belum memenuhi target bahkan bukan total pendapatan tertinggi. Maka dari itu values yang digunakan adalah values dengan unit tertinggi, selain itu pemilihan values tersebut supaya tidak terjadi kekurangan stok.

DAFTAR PUSTAKA

Purwanto, E. (2020). *Pengantar Bisnis Era Revolusi Industri 4.0*. Purwokerto: SASANTI INSTITUTE.

LAMPIRAN

Lampiran 1. Data Sales

Customer_ID	Product_ID	Units	Tanggal
IND76X	G32N	64	01/01/2021
KUS50X	J22S	26	02/01/2021
AND23X	K78A	52	03/01/2021
IKA40X	M90L	13	04/01/2021
RAT11X	R28K	54	05/01/2021
KUS50X	R28K	30	06/01/2021
IKA40X	R28K	33	07/01/2021
AHM07X	G32N	56	08/01/2021
IND76X	K78A	39	09/01/2021
AHM07X	G32N	56	10/01/2021
IND76X	R28K	8	11/01/2021
RAT11X	R28K	24	12/01/2021
AHM07X	G32N	43	13/01/2021
IND76X	J22S	25	14/01/2021
IKA40X	K78A	63	15/01/2021
RAT11X	K78A	46	16/01/2021
RAT11X	R28K	21	17/01/2021
AND23X	M90L	36	18/01/2021
AND23X	K78A	31	19/01/2021
KUS50X	J22S	7	20/01/2021
RAT11X	R28K	12	21/01/2021
RAT11X	G32N	44	22/01/2021
RAT11X	K78A	34	23/01/2021
IND76X	R28K	6	24/01/2021
IND76X	R28K	58	25/01/2021
KUS50X	G32N	10	26/01/2021
IKA40X	R28K	38	27/01/2021
AHM07X	M90L	12	28/01/2021
RAT11X	K78A	13	29/01/2021
RAT11X	G32N	44	30/01/2021
IKA40X	J22S	57	31/01/2021
IKA40X	M90L	53	01/02/2021
IKA40X	J22S	40	02/02/2021
KUS50X	G32N	61	03/02/2021
IKA40X	M90L	61	04/02/2021
KUS50X	K78A	59	05/02/2021
RAT11X	G32N	14	06/02/2021
IND76X	R28K	23	07/02/2021

KUS50X	G32N	16	08/02/2021
AHM07X	J22S	24	09/02/2021
AND23X	G32N	65	10/02/2021
RAT11X	K78A	38	11/02/2021
AHM07X	J22S	7	12/02/2021
IND76X	G32N	16	13/02/2021
IKA40X	M90L	23	14/02/2021
AND23X	R28K	8	15/02/2021
AHM07X	K78A	10	16/02/2021
AHM07X	M90L	27	17/02/2021
AND23X	M90L	17	18/02/2021
IND76X	K78A	18	19/02/2021
KUS50X	G32N	58	20/02/2021
IKA40X	J22S	59	21/02/2021
KUS50X	R28K	62	22/02/2021
IND76X	J22S	35	23/02/2021
IND76X	M90L	22	24/02/2021
AND23X	G32N	42	25/02/2021
IKA40X	J22S	19	26/02/2021
IKA40X	J22S	40	27/02/2021
KUS50X	M90L	62	28/02/2021
AHM07X	J22S	9	01/03/2021
RAT11X	K78A	12	02/03/2021
KUS50X	K78A	20	03/03/2021
IKA40X	M90L	57	04/03/2021
AND23X	J22S	6	05/03/2021
IND76X	J22S	35	06/03/2021
RAT11X	R28K	59	07/03/2021
IKA40X	G32N	22	08/03/2021
IKA40X	J22S	61	09/03/2021
AND23X	R28K	42	10/03/2021
IKA40X	G32N	30	11/03/2021
RAT11X	K78A	60	12/03/2021
RAT11X	G32N	62	13/03/2021
KUS50X	J22S	17	14/03/2021
KUS50X	R28K	11	15/03/2021
AND23X	J22S	11	16/03/2021
IND76X	J22S	15	17/03/2021
AND23X	G32N	49	18/03/2021
IKA40X	R28K	23	19/03/2021
AND23X	G32N	17	20/03/2021
RAT11X	J22S	20	21/03/2021
IND76X	M90L	44	22/03/2021
AHM07X	G32N	63	23/03/2021
AHM07X	M90L	63	24/03/2021

AND23X	K78A	15	25/03/2021
AND23X	R28K	40	26/03/2021
IND76X	M90L	47	27/03/2021
AND23X	R28K	64	28/03/2021
KUS50X	R28K	49	29/03/2021
IND76X	J22S	56	30/03/2021
AHM07X	J22S	21	31/03/2021
IND76X	G32N	25	01/04/2021
AHM07X	G32N	10	02/04/2021
RAT11X	G32N	29	03/04/2021
AND23X	K78A	53	04/04/2021
AHM07X	R28K	49	05/04/2021
IND76X	M90L	29	06/04/2021
AHM07X	M90L	19	07/04/2021
AND23X	G32N	17	08/04/2021
IKA40X	J22S	53	09/04/2021
AND23X	K78A	29	10/04/2021
AND23X	J22S	16	11/04/2021
AND23X	J22S	27	12/04/2021
AHM07X	J22S	10	13/04/2021
KUS50X	J22S	50	14/04/2021
AND23X	R28K	51	15/04/2021
IKA40X	J22S	61	16/04/2021
AND23X	R28K	42	17/04/2021
IKA40X	G32N	30	18/04/2021
RAT11X	K78A	60	19/04/2021
RAT11X	G32N	62	20/04/2021
KUS50X	J22S	17	21/04/2021
KUS50X	R28K	11	22/04/2021
AND23X	J22S	11	23/04/2021
IND76X	J22S	15	24/04/2021
AND23X	G32N	49	25/04/2021
IKA40X	R28K	23	26/04/2021
AND23X	G32N	17	27/04/2021
RAT11X	J22S	20	28/04/2021
IND76X	M90L	44	29/04/2021
AHM07X	G32N	63	30/04/2021
AHM07X	M90L	63	01/05/2021
AND23X	K78A	15	02/05/2021
AND23X	R28K	40	03/05/2021
IND76X	M90L	47	04/05/2021
AND23X	R28K	64	05/05/2021
KUS50X	R28K	49	06/05/2021
IND76X	J22S	56	07/05/2021
AHM07X	J22S	21	08/05/2021

IND76X	G32N	25	09/05/2021
AHM07X	G32N	10	10/05/2021
RAT11X	G32N	29	11/05/2021
AND23X	K78A	53	12/05/2021
AHM07X	R28K	49	13/05/2021
IND76X	M90L	29	14/05/2021
AHM07X	M90L	19	15/05/2021
AND23X	G32N	17	16/05/2021
IKA40X	J22S	53	17/05/2021
AND23X	K78A	29	18/05/2021
AND23X	J22S	16	19/05/2021
AND23X	J22S	27	20/05/2021
AHM07X	J22S	10	21/05/2021
KUS50X	J22S	50	22/05/2021
AND23X	R28K	51	23/05/2021

Lampiran 2. Data Customer

Customer_ID	Name	Address
IND76X	Indra Agustina	Jawa
KUS50X	Kusuma Dina	Kalimantan
AND23X	Andi Ade	Sulawesi
IKA40X	Ika Indah	Jawa
RAT11X	Ratna Fitria	Sumatra
AHM07X	Ahmad Surya	Bali

Lampiran 3. Data Produk

Product_id	Product	Price
G32N	Gaun	400,000
J22S	Jas	350,000
K78A	Kemeja	250,000
M90L	Mantel	170,000
R28K	Rok	85,000