

BAB II

LANDASAN TEORI

2.1 Definisi Sistem

Mempelajari suatu sistem akan lebih mengenal bila mengetahui terlebih dahulu apakah sistem itu. Pengertian tentang sistem pertama kali dapat diperoleh dari definisi sistem itu sendiri. Jika kita perhatikan seksama, diri kita juga terdiri dari berbagai sistem untuk mengantar kita kepada tujuan hidup kita. “Azhar Susanto (2013:22) dalam bukunya yang berjudul Sistem Informasi Akuntansi, Sistem merupakan kumpulan atau grup dari sub *system* atau bagian atau komponen apapun baik fisik ataupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu”.

Berdasarkan pengertian sistem dari beberapa ahli diatas, dapat disimpulkan bahwa sistem adalah kumpulan elemen-elemen yang memiliki tujuan tertentu dengan menghasilkan sebuah output yang teratur

Dari definisi ini dapat dirinci lebih lanjut pengertian sistem secara umum, yaitu sebagai berikut:

1. Setiap sistem terdiri berbagai unsur. Sistem pernafasan kita terdiri dari suatu kelompok unsur, terdiri dari hidung, saluran pernafasan, paru-paru dan darah.
2. Unsur-unsur tersebut merupakan bagian yang tak terpisahkan dari sistem yang bersangkutan.
3. Unsur-unsur di dalam sistem tersebut bekerja sama untuk mencapai tujuan sistem.
4. Suatu sistem merupakan bagian dari sistem lain yang lebih besar.

Suatu sistem dapat dirumuskan sebagai setiap kumpulan komponen atau subsistem yang dirancang untuk untuk mencapai suatu tujuan. Dengan pendekatan sistem kita berhubungan dengan komponen perseorangan, dan kita lebih menekankan perannya di dalam sistem dari pada perannya sebagai suatu keseluruhan individu.

Marshall B.Romney dan Paul John Steinbart (2014:3), “Sistem adalah serangkaian dua atau lebih komponen yang saling terkait dan berinteraksi untuk mencapai tujuan”.

2.1.1 Karakteristik Sistem

Model umum sebuah sistem terdiri dari input, proses dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Selain itu sebuah sistem juga memiliki karakteristik atau sifat-sifat yang tertentu, yang mendirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut:

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan.

Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar yang disebut dengan supra sistem.

2. Batasan sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antar sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan yang tidak dapat dipisah-pisah.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem yang menguntungkan merupakan energi bagi sistem tersebut. Lingkungan luar sistem yang merugikan harus dikendalikan karena kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung Sistem (*Interface*)

Sebagai media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau interface. Penghubung memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Keluaran subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung. Dengan demikian terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Sebagai contoh di dalam suatu unit sistem komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputer sementara data adalah *signal input* yang akan diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

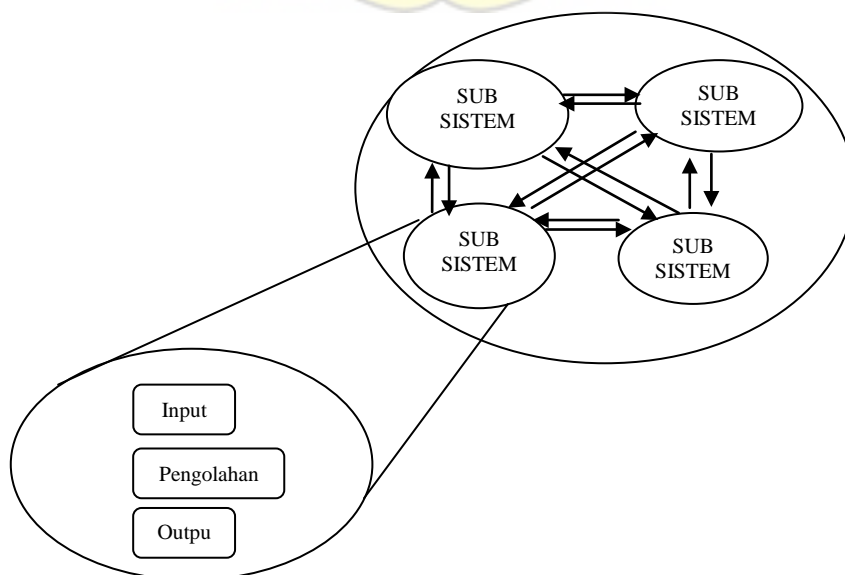
Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain. Seperti contoh sistem informasi keluaran yang dihasilkan adalah informasi, yang mana informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang merupakan input bagi subsistem lainnya.

7. Pengolah Sistem (*Proses*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Sebagai contoh sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deternistik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya, suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan lihat pada gambar 2.1.



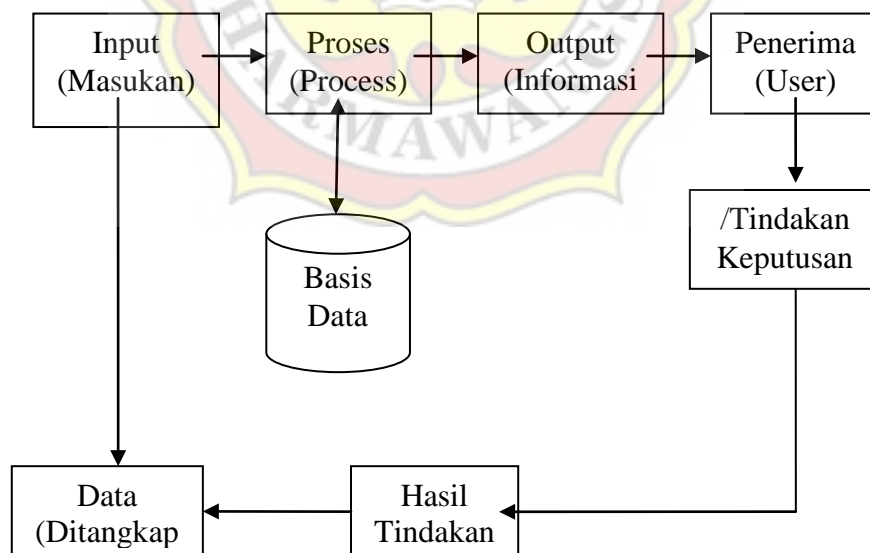
Gambar 2.1. Karakteristik Sistem

2.2. Definisi Informasi

Menurut Dr. Deni Darmawan dan Kunkun Nur Fauzi (2013:2), “Informasi merupakan hasil dari pengolahan data, akan tetapi tidak semua hasil dari pengolahan data yang tidak dari pengolahan tersebut bisa disebut informasi, hasil pengolahan data yang tidak memberikan makna, arti atau penjelasan serta tidak bermanfaat bagi seseorang atau penerima informasi bukanlah merupakan informasi bagi orang tersebut”.

Gordon B. Davis dalam bukunya Bambang Hartono (2013:15), “informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berguna dan nyata atau berupa nilai yang dapat dipahami oleh penerima informasi sehingga dapat digunakan menjadi sebuah keputusan sekarang maupun yang akan datang”.

Artinya dengan adanya informasi, data yang dibentuk menjadi lebih berarti bagi penerimanya lihat pada gambar 2.2.



Gambar 2.2. Siklus Informasi

2.2.1 Kualitas Informasi

Informasi bisa dikatakan berkualitas, jika memiliki delapan kriteria

menurut Bambang Hartono dalam bukunya yang berjudul Sistem Informasi Manajemen Berbasis Komputer (2013:17-18), berikut diantaranya :

1. Relevansi, berarti informasi disediakan atau disajikan untuk digunakan. Oleh karena itu, informasi yang bernilai tinggi adalah yang relevan dengan kebutuhan yaitu untuk apa informasi itu akan digunakan.
2. Kelengkapan dan Keluasan, informasi akan bernilai semakin tinggi, jika tersaji secara lengkap, teliti dan memiliki penjelasan atau arti dalam cakupan yang luas. Informasi yang sepotong-potong, penjelasan yang tidak jelas serta tidak tersusun sistematis tentu tidak akan banyak artinya dan tidak dapat disebut sebagai informasi.
3. Kebenaran, kebenaran informasi ditentukan oleh valid atau tidaknya dan dapat dibuktikan secara nyata. Informasi yang bernilai tinggi adalah informasi yang benar-benar berasal dari fakta bukan berdasarkan opini seseorang.
4. Terukur, informasi berasal dari sebuah hasil pengukuran dan pencatatan terhadap fakta, sehingga bisa dikatakan informasi jika informasi yang dilacak kembali datanya, data tersebut dapat diukur sesuai dengan faktanya.
5. Keakuratan, informasi berasal dari hasil pengukuran dan pencatatan terhadap fakta.
6. Kejelasan, informasi dapat disajikan dalam berbagai bentuk teks, table, grafik, chart dan lain-lain. Namun, apapun bentuk yang dipilih, yang penting adalah menjadikan pemakai mudah memahami maknanya. Oleh sebab itu, selain bentuk penyajiannya harus benar juga harus diperhatikan kemampuan pemakai dalam memahaminya.

7. Keluwesan, informasi yang dapat diubah-ubah bentuk penyajiannya sesuai dengan kebutuhan dan situasi yang dihadapi.
8. Ketepatan Waktu, informasi yang baik adalah informasi yang disajikan tepat pada saat dibutuhkan.

2.2.2 Nilai Informasi

Suatu informasi memiliki nilai karena informasi tersebut dapat dijadikan keputusan yang baik serta menguntungkan bagi pemakai informasi tersebut. Besarnya nilai informasi yang tepat dapat didapatkan dari perbedaan hasil yang didapat dari keputusan yang baru dengan hasil keputusan yang lama dikurangi dengan biaya untuk mendapatkan informasi tersebut. Menurut Gordon B. Davis dalam buku Bambang Hartono 2013 dengan judul Sistem Informasi Manajemen Berbasis Komputer, nilai informasi dikatakan sempurna jika terdapat perbedaan antara kebijakan optimal, tanpa informasi yang sempurna dan kebijakan optimal maka informasi tersebut tidak dapat dinyatakan dengan jelas.

Nilai suatu informasi dapat ditentukan berdasarkan sifatnya. Tentang 10 sifat yang dapat menentukan nilai informasi. Yaitu sebagai berikut :

1. Kemudahan dalam memperoleh, diperoleh secara mudah berbeda jika informasi yang penting dan dibutuhkan menjadi tidak bernilai jika sulit diperoleh.
2. Sifat luas dan kelengkapannya, mempunyai lingkup atau cakupan yang luas dan lengkap. Informasi yang tidak lengkap menjadi tidak bernilai, karena tidak dapat digunakan.
3. Ketelitian (accuracy), mempunyai ketelitian yang tinggi atau akurat, informasi menjadi tidak bernilai jika tidak akurat, karena akan

mengakibatkan kesalahan dalam mengambil keputusan.

4. Kecocokan dengan pengguna (relevance), Sesuai dengan kebutuhan penggunanya, jika informasi tidak sesuai dengan kebutuhan penggunanya, maka tidak dapat dimanfaatkan oleh pengguna informasi.
5. Ketepatan waktu, diterima oleh pengguna pada saat yang tepat. Jika terlambat diterima atau using, maka tidak dapat dimanfaatkan pada saat pengambilan keputusan.
6. Kejelasan (clarity), informasi yang jelas akan meningkatkan kesempurnaan nilai informasi.
7. Fleksibilitas, memiliki fleksibilitas tinggi, fleksibilitas informasi diperlukan oleh para manajer atau pimpinan atau selain pejabat pada saat pengambilan keputusan.
8. Dapat dibuktikan, dapat dibuktikan kebenarannya, kebenaran informasi bergantung pada validitas data sumber yang diolah.
9. Tidak ada prasangka, apabila informasi tersebut tidak menimbulkan prasangka dan keraguan adanya kesalahan informasi.
10. Dapat diukur, informasi untuk pengambilan keputusan seharusnya dapat diukur agar dapat mencapai nilai yang sempurna.

2.3. Definisi Sistem Informasi

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan. (Tata Sutabri, 2012).

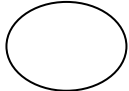
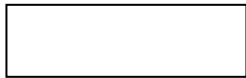

Menurut Sutarman (2012:13), “Sistem informasi adalah ”Sistem dapat didefinisikan dengan mengumpulkan, memproses, menyimpan, menganalisis, menyebarkan informasi untuk tujuan tertentu. Seperti sistem lainnya, sebuah sistem informasi terdiri atas input (data, instruksi) dan output (laporan, kalkulasi)”. Dari beberapa pendapat yang dikemukakan di atas dapat diambil kesimpulan bahwa “Sistem informasi adalah sebuah sistem yang terdiri dari pengumpulan, pemasukan, pemrosesan data, penyimpanan, pengolahan, pengendalian dan pelaporan sehingga tercapai sebuah informasi yang mendukung pengambilan keputusan didalam suatu organisasi untuk dapat mencapai sasaran dan tujuannya”.

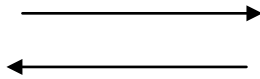
2.4. Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah sebuah gambaran untuk menentukan dan memberikan arah alur data yang berbentuk diagram yang menggunakan lambang-lambang atau simbol-simbol untuk menggambarkan arus proses data dari suatu sistem yang terstruktur.

Berikut adalah tabel 2.1 simbol - simbol DFD :

Tabel 2.1. Simbol DFD

Simbol	Fungsi
	Proses, digunakan untuk menggambarkan sebuah sistem atau proses.
	<i>External Entity</i> , digunakan untuk menggambarkan sebuah objek.
	<i>Storage</i> , digunakan untuk menggambarkan/menunjukkan sebuah file data.




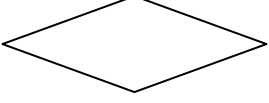

	Arus data, digunakan untuk menggambarkan sebuah arus data.
---	--

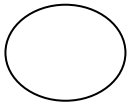
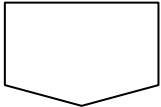
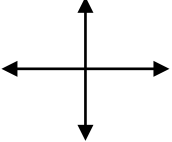



2.5. Flowchart

Flowchart atau bagan alir adalah suatu skema yang menggambarkan urutan kegiatan dari suatu program dari awal sampai akhir. Tujuan utama dari penggunaan *Flowchart* ini adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, teratur, rapi dan jelas dengan menggunakan simbol-simbol yang standart.

Berikut ini adalah gambar 2.2 dari simbol-simbol standar yang digunakan pada program *Flowchart*.

Tabel 2.2.Flowchart

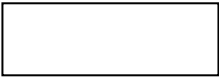
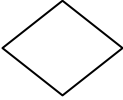
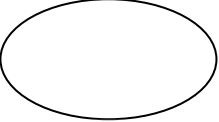

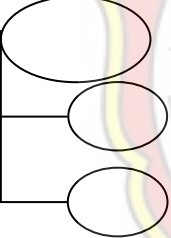
No	Simbol	Fungsi
1		Terminal, untuk memulai atau mengakhiri suatu program.
2		Proses, suatu simbol yang menunjukkan setiap pengolahan yang dilakukan oleh komputer.
3		<i>Input/Output</i> untuk memasukkan data atau menunjukkan hasil dari suatu proses.
4		<i>Decision</i> , suatu kondisi yang menghasilkan dua kemungkinan jawaban atau pilihan.
5		<i>Predefined Proses</i> , Untuk menyatakan sekumpulan langkah-langkah proses yang ditulis sebagai prosedur.

6		<i>Connector</i> , satu prosedur akan masuk/keluar melalui simbol ini dalam lembar yang sama.
7		<i>Off-line, Connector</i> , merupakan simbol masuk atau keluarnya suatu prosedur pada lembar kerja lainnya.
8		Arus/ <i>Flow</i> dari prosedur yang dapat dilakukan dari atas ke bawah, dari bawah ke atas, dari kiri ke kanan atau dari kanan ke kiri.
9		<i>Document</i> , merupakan simbol untuk data yang berbentuk kertas maupun untuk informasi.
10		Simbol untuk <i>output</i> yang ditujukan ke suatu device, seperti printer, plotters dan lain sebagainya.
11		Media penyimpanan drum, merupakan media penyimpanan secara tetap.

2.6. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. ERD merupakan model jaringan data yang menekankan pada struktur – struktur dan relation data. ERD merupakan alat untuk pembuatan model data secara grafik, maka ERD memiliki simbol–simbol yang digunakan untuk menggambarkan model data seperti terdapat pada table berikut :

Tabel 2.3 Simbol – simbol Entity Relationship Diagram

No	Simbol	Arti	Definisi
1		Entitas	Suatu kumpulan objek atau sesuatu yang dapat dibedakan atau didefinisikan secara unik.
2		<i>Relationship</i>	Hubungan yang terjadi antara satu atau lebih entity.
3		Atribut	Karakteristik dari entity atau relationship yang menyediakan penjelasan detail tentang entity atau <i>relationship</i> tersebut.
4		Atribut Kunci	Atribut yang digunakan untuk menentukan suatu entity secara unik.
5		Atribut Komposisi	Atribut yang memiliki sekelompok nilai untuk setiap instansi <i>entity</i> .

1. Entitas dan atribut.

Seperti telah dijelaskan di atas, entitas adalah tempat penyimpanan data, maka entitas yang digambarkan dalam ERD ini merupakan *data store* yang ada di DFD dan akan menjadi *file* data di komputer. Entitas adalah suatu objek dan memiliki nama. Secara sederhana dapat dikatakan bahwa jika objek ini tidak ada di suatu *enterprise* (lingkungan tertentu), maka *enterprise* tersebut tidak dapat berjalan normal.

2. Relasi

Relasi adalah penghubung antara satu entitas (*master file*) dengan entitas lain

di dalam sebuah sistem komputer. Pada akhirnya, relasi akan menjadi *file* transaksi (*transaction file*) di computer.

3. Derajat Kardinalitas (*Cardinality Degree*)

Hubungan antar entitas ditandai pula oleh derajat kardinalitas. Fungsi dari derajat kardinalitas ini adalah untuk menentukan entitas kuat dan entitas lemah. Tiga jenis derajat kardinalitas adalah :

- a. *One to one*, dilambangkan dengan 1 : 1.
- b. *One to many*, dan sebaliknya, yang dilambangkan dengan 1 : M dan sebaliknya.
- c. *Many to many*, dilambangkan dengan M : M atau M : N.

4. Penentuan *Primary Key*

Disetiap entitas di dalam ERD (di gambar 12 di atas), seharusnya ada atribut (*field*) yang dipilih untuk dijadikan kunci utama atribut (*primary key/ key field*), yaitu atribut yang dijadikan identitas yang menjamin keunikan (tidak ada yang sama) isi datanya.

2.7 Pengertian Penjualan

Istilah penjualan mempunyai pengertian dalam arti mikro dan pengertian dalam arti makro. Pengertian penjualan dalam arti mikro yaitu penyelenggaraan kegiatan yang berusaha mencapai tujuan organisasi, dengan cara memperkirakan kebutuhan langganan dan mengarahkan suatu harus barang dan jasa untuk memenuhi kebutuhan dari produsen ke konsumen, sedangkan dalam arti makro penjualan merupakan proses sosial yang mengarahkan arus barang-barang dan jasa-jasa dari suatu perekonomian dari produsen ke konsumen, dengan cara yang seefektif menyesuaikan penawaran dan permintaan dan untuk mencapai tujuan yang diinginkan masyarakat. Penjualan merupakan kegiatan manusia yang

bertujuan untuk memuaskan kebutuhan dan keinginan langganan, melalui proses pertukaran dan kepada pihak-pihak yang berkepentingan dengan perusahaan.

Penjualan merupakan kegiatan yang dilakukan oleh penjual dalam menjual barang atau jasa dengan harapan akan memperoleh laba dari adanya transaksi-transaksi tersebut (Mulyadi, 2014:57)

Menurut Basu Swastha dalam Irwan Sahaja (2014: 33), penjualan adalah suatu proses pertukaran barang atau jasa antara penjual dan pembeli.

2.7.1 Tahap-Tahap Penjualan

Dalam kegiatan penjualan, ada tahap yang ditempuh oleh penjual agar tujuan penjualan dapat tercapai. Menurut Basu Swasta (2014:121), yaitu :

1. Persiapan Sebelum Penjualan

Disini kegiatan dilakukan dengan mempersiapkan tenaga penjual dengan memberikan pengertian tentang barang yang akan dijual, pasar yang dituju, dan teknik-teknik penjualan yang harus dilakukan. Selain itu perlu juga pemberian pengetahuan mengenai motivasi dan perilaku pembeli dalam segmen pasar yang dituju.

2. Penentuan Lokasi Pembeli Potensial

Dengan menggunakan data pembelian tahun lalu maupun sekarang, penjual dapat menentukan karakteristik calon pembeli potensial. Oleh karena itu, pada tahap yang ditentukan lokasi dari segmen pasar yang menjadi sasarannya.

3. Pendekatan Pendahuluan

Sebelum melakukan penjualan harus dipelajari semua data yang berhubungan dengan lokasi pasar dan kemungkinan persaingan di lokasi. Juga

penting untuk mengenalkan produk lebih awal lewat promosi di tempat-tempat yang telah ditentukan sebagai tempat potensial untuk menjual produk.

4. Melakukan Penjualan

Dalam tahap ini penjual telah sampai pada penjualan produk pada segmen pasar yang telah ditetapkan. Penjualan harus memperhatikan permintaan pasar akan produk karena seringkali terjadi permintaan mendadak yang tidak dapat dipenuhi oleh penjual sehingga mengakibatkan kepercayaan konsumen terhadap produk mengalami penurunan.

5. Pelayanan Purna Jual

Setelah produk dijual dan diterima pembeli atau konsumen tahap selanjutnya adalah pelayanan purna jual dimana penjual mempersiapkan penerimaan kembali barang-barang yang cacat atau rusak, mendengar keluhan konsumen tentang kelemahan produk yang dibelinya dan melayani penggantian kembali barang-barang yang rusak itu sesuai dengan jaminan (garansi) produk yang diberikan, pemberian jasa reparasi, latihan tenaga operasional dan cara penggunaannya, serta pengantaran barang kerumah.

2.7.2 Tujuan Penjualan

Pada umumnya, perusahaan mempunyai tiga tujuan umum dalam penjualannya seperti yang dikemukakan oleh Basu Swasta (2014:80) :

1. Berusaha mencapai volume penjualan tertentu.
2. Berusaha mencapai laba atau profit akhir melebihi biaya yang dikeluarkan.
3. Menunjang pertumbuhan perusahaan.

2.8 *Visual Basic.Net 2008*

Microsoft Visual.Net 2008 adalah sebuah aplikasi yang digunakan sebagai alat bantu di dalam membangun suatu program aplikasi yang berbasis komputer. *Visual Studio.Net 2008* merupakan generasi yang terbaru yang dikeluarkan *Microsoft* dimana sebelumnya ada *Visual Studio.Net 2003* & *Visual Studio.Net 2008* adalah, sebagai berikut :

1. Memahami *platform Microsoft.NET*
2. Mengenal *Intergrated Development Environment (IDE) Visual Basic NET*
3. Memahami stuktur *project Visual Basic.NET*
4. Memahami jenis – jenis aplikasi.NET
5. Mampu membangun *Graphical User Interface (GUI)* menggunakan *Visual Basic.NET*

2.8.1 *Dasar Teori Visual Studio.Net 2008*

Platfoem.NET Framework Microsoft.NET yang awalnya disebut *Next Generation Windows.Service (NGWS)* adalah suatu *platform* untuk membangun dan menjalankan generasi penerus aplikasi – aplikasi terdistribusi. *Microsoft.NET* merupakan *framework* (kerangka) pengembangan yang menyediakan antarmuka pemrograman baru untuk layanan *Windows* dan *API (Application Programming Interface)*.

Microsoft.NET merupakan strategi *Microsoft* untuk menghubungkan sistem, informasi, ada alat (*device*), sehingga orang dapat berkomunikasi serta berkolaborasi dengan lebih efektif. *Teknologi.NET* terintegrasi penuh melalui produk – produk *Microsoft*, dan meyediakan kemampuan untuk mengembangkan solusi dengan menggunakan *Web service*.

Platform Microsoft.NET terdiri dari lima komponen utama yang tersusun dalam tiga lapisan (*layer*). Lapisan paling bawah adalah sistem operasi, lapisan kedua terdiri dari tiga komponen, lapisan teratas adalah *Visual Studio.Net*.

Microsoft Visual Studio.Net merupakan kumpulan lengkap *tools* pengembangan untuk membangun aplikasi web asp.net, sml, web Service, aplikasi desktop dan aplikasi mobile. Di dalam *Visual Studio* inilah bahasa – bahasa pemograman.NET seperti *Visual Basic*, Visual C++, Visual C# (C Sharp), dan Visual C# (Gsharp) semuanya menggunakan lingkunag pengembangan terintergrasi atau ID yang sama sehingga memungkinkan untuk saling berbagi tools dan fasilitas.

2.8.2 Fitur Baru *Visual Studio.Net* 2008

Berikut ini akan dijelaskan mengenai fitur – fitur baru yang ada pada *Visual Studio. NET* 2008 :

1. Dukungan terhadap *Linq*

Linq merupakan suatu fitur baru yang dikeluarkan oleh Microsoft khususnya pada bahasa pemograman C#. *Linq* sebenarnya adalah sekumpulan operator – operator *query standart* yang memanpukan kita untuk bekerja dengan data yang ada dalam database. Teknologi yang digunakan oleh *Linq* di dalam pemanipulasian data adalah menggunakan teknologi orm (*Object Relational Mapping*) bukan teknologi konvensional yaitu RDBMS. Konsep dari teknologi ORM adalah menjadikan semua tabel yang terlibat dalam suatu aplikasi menjadi sebuah objek dimana properti – properti dari objek tersebut didapat dari kolom – kolom yang dimiliki oleh suatu tabel. Keuntungan dari *Linq* ini adalah kita dapat

melakukan pengecekan *query* pada saat kompilasi program dilakukan dan kemampuan untuk mendebuq *query* langkah perlangkah.

2. Dukungan Terhadap *Expression Blend*

Expression Blend adalah aplikasi yang digunakan untuk membantu XAML untuk aplikasi *silverlight*. Kita dapat menginstall *Expression Blend* sebagai *plug-in* di dalam *Visual Studio.Net* 2008 dengan menjadikan *Expression Blend* sebagai *plug-in* akan membantu kita dalam desain *website* dan pemakaian *javascript*.

3. *Windows Presentation Foundation*

WPF banyak menyediakan fungsi grafis. *Visual Studio.Net* 2008 banyak memiliki *library Windows Presentation Foundation* dimana dengan fitur ini maka kita akan sangat terbantu di dalam membangun aplikasi grafis 2 dimensi (2D) dan 3 dimensi (3D).

4. Dukungan *Multi Framework*

Visual Studio.Net 2008 mendukung banyak *framework*. Dalam artian dengan menggunakan *Visual Studio.Net* 2008 kita dapat melakukan kompilasi untuk *framework* 2.0, 3.0 dan 3.5 bahkan *framework* 1.1.

5. Dukungan Terhadap *Ajax* Dalam ASP.NET

Pada seri sebelumnya, untuk menggunakan *ajax* kita harus instal terlebih dahulu secara terpisah tidak *built-in* di dalam *Visual Studio.Net* nya. Namun dengan *Visual Studio.Net* 2008, *library ajax* akan secara otomatis terinstall ketika kita menginstall *Visual Studio.Net* 2008.

6. Dukungan Terhadap *Debugging* untuk *Javascript*

Sejak munculnya *javascript* di dalam pembangunan sebuah *website*, pada *developer* mengalami kebingungan di dalam mencari kesalahan yang ada

di dalam *javascript*. Namun di dalam *Visual Studio.Net* 2008 kita dapat meakukan debugging terhadap kesalahan-kesalahan yang mungkin terjadi di dalam *javascript*.

7. Dukungan Terhadap *Nested Masterpage*

Sebenarnya *Nested Masterpage* sudah ada pada *Visual Studio.Net* 2005 namun ada permasalahan di dalam *Visual Studio.Net* 2005 dimana *Nested Masterpage* tidak dapat diedit dalam *mode designer* atau dalam kondisi *wysiwyg*. Namun dalam *Visual Studio.Net* 2008, pengeditan *nested masterpage* dapat dilakukan.

8. Dukungan Terhadap *linq Intellisense* dan *javascript Intellisense* Pada Aplikasi *Silverlight*.

Javascript Intellisense memungkinkan pada *developer* tidak perlu lagi untuk mengingat-ingat *syntax* ataupun fungsi-fungsi yang ada di dalam *javascript* karena *Visual Studio*.

Dengan adanya fitur ini maka pembuatan *script client* menggunakan *javascript* akan lebih mudah termasuk dalam aplikasi *ajax* dan *Silverlight*.

9. Pengorganisasian Penggunaan *Keyword Using* dan *Import*

Dalam *Visual Studio.Net* 2008 pengorganisasian *keyword using* dan *import* menjadi lebih efisien dikarenakan *Visual Studio.Net* 2008 akan secara otomatis menghapus *namespace* yang tidak digunakan. Cara yang dilakukan dengan *select* seluruh *namespace*, klik kanan dan pilih *Remove Unused Usings*.

10. *Intellisense Filtering*

Dalam Versi *Visual Studio.Net* 2008, misalkan *Visual Studio.Net* 2005, ketika kita mengetikkan suatu huruf atau karakter lainnya kedalam kotak

intellisense maka semua *item* yang berhubungan dengan huruf atau karakter tersebut akan dimunculkan. Sebagai contoh, ketika kita menuliskan 'a'. maka seluruh *item* yang memiliki huruf a baik di depan, tengah ataupun belakang akan dimunculkan. Namun dalam *Visual Studio.Net 2008* semua *item* yang diawali dengan huruf 'a' saja yang akan dimunculkan.

11. Posisi Dari Kotak *Intellisense*

Pada versi sebelumnya ketika kita mengetikkan suatu perintah dan diakhiri dengan tanda titik ('.') maka kotak *intellisense* akan muncul. Namun terkadang kemunculan kotak *intellisense* ini mengganggu kita, misalkan fungsi atau perintah yang ada dibawah perintah atau fungsi yang memunculkan kotak *Intellisense* tertutupi sehingga kita tidak dapat mengetahui fungsi atau perintah tersebut. Namun dalam *Visual Studio.Net 2008* hal ini dapat diatas dengan menekan tombol Ctrl. Dengan menekan tombol Ctrl maka kita membuat kotak tersebut jadi transparan. Dengan transparannya kotak *intellisense* ini maka perintah atau fungsi yang tertutupi jadi kelihatan lagi.

12. *Split View* Pada *Visual Studio.Net 2008*

Pada versi sebelumnya, tampilan *Visual Studio.Net* juga dapat dibagi atau displit. Namun split yang terjadi hanya secara horizontal. Namun pada versi *Visual Studio.Net 2008* kita dapat membagi tampilan *windows* secara vertical. Fitur ini memungkinkan *developer* untuk memanfaatkan layar monitor mereka yang lebar secara maksimum. Kemudian fitur yang menarik lainnya adalah apa yang kita *select* pada bagian *source code* akan juga secara otomatis *menselect item* yang sesuai dengan *code* pada

windows desain.

13. HTML *Javascript Warning*, Bukan Error

Pada *Visual Studio.Net* 2005, error antara html dengan C# atau VB.Net digabungkan dalam 1 *windows*. Namun pada *Visual Studio.Net* 2008, terjadi pemisahan dan juga error yang terjadi dari html dan *javascript* hanya dianggap *warning* bukan error. Fitur ini tidak mutlak dalam artian kita bisa membuat kesalahan pada html atau *Javascript* merupakan suatu error bukan *warning*.

14. Kemampuan *Debugging Source Code Library.Net Framework*

Dalam *Visual Studio.Net* 2008, kita dapat melakukan *debug* terhadap kode-kode yang ada dalam *library.Net Framework* baik *method* ataupun properti. Misalkan kita ingin *mendebug method* *Databind* dari kontrol *dataGrid*. Maka yang kita lakukan adalah meletakkan pointer *debugging* pada *method Databind* yang lakukan *debug* adalah dengan menggunakan tombol F10 atau F11.

15. *Library Silverlight* sudah *Built-in*

Pada versi sebelumnya, untuk dapat membangun aplikasi *silverlight* maka kita harus melakukan instalasi *Silverlight* SDK secara terpisah. Namun dalam *Visual Studio.Net* 2008 maka kita *silverlight* adalah *built-in* dalam artian sudah terinstall otomatis bersamaan terinstallnya *Visual Studio.Net* 2008.

16. *Visual Studio LINQ Designer*

Dalam *Visual Studio.Net* 2008, kita memiliki *Linq Designer* dimana kita dapat mendesain secara visual keterhubungan antara tabel yang ada dalam satu database khususnya untuk *linq-to-sql*.

17. C++SDK *Built-in*

Pada versi sebelumnya, sangat sulit untuk *mendownload* dan melakukan konfigurasi terhadap *library* dan *tool-tool* yang berhubungan dengan c++ sdk dalam membangun aplikasi *windows*. Namun dalam *Visual Studio.Net 2008*, *library* dan *tool-tool* ini sudah *built-in*.

18. Arsitektur MUI (*Multilingual User Interface*)

MUI mengandung paket *library-library* dari *Microsoft Windows* sampai kepada *Microsoft Office*. Hal ini mendukung pengguna untuk mengubah tampilan text bahasa sesuai dengan keinginan mereka. Saat ini 9 bahasa yang didukung yaitu antar lain *English, Spanish, French, German, Italian, Chinese Simplified Chinese Traditional, Japanese dan Korean*.

19. Dukungan Terhadap *Microsoft Popfly*

Eksplorer dari *Microsoft Popfly* bersifat *add-on* dalam *Visual Studio.Net 2008*. Dengan *add-on* ini kita dapat secara langsung melakukan *hosting* terhadap aplikasi *silverlight* dan *object Marshup*.

20. *Tool dan Ebook* Gratisan

Untuk mendukung pemakaian *Visual Studio.Net 2008* maka *Microsoft* menyediakan versi gratisan seperti *Visual Studio.Net Express, Visual C# Express, dll*. Selain itu banyak juga *ebook-ebook* yang berhubungan dengan *Visual Studio.Net 2008* yang bisa *download* secara gratis.

2.9 *Crystal Report 8.5*

Crystal Report adalah intelijen bisnis aplikasi yang digunakan untuk merancang dan menghasilkan dari berbagai sumber data. Beberapa aplikasi lain, seperti *Microsoft Visual Studio*, bundel yang OEM *Crystal Report* Versi 8.5

sebagai alat pelaporan tujuan utama. *Crystal Report* menjadi standar *de facto* laporan penulisan ketika *Microsoft* merilis dengan *Visual Studio 8.5*

Crystal Report dirancang untuk membuat laporan yang dapat digunakan dengan bahasa pemrograman berbasis *Windows*, seperti *Boarland Delphi Visual Basic*, *Visual C/C++*, dan visual interdev. Beberapa kelebihan dari *Crystal Report* ini adalah :

1. Dari segi pembuatan laporan, tidak terlalu rumit yang memungkinkan para *programmer* pemula sekalipun dapat membuat laporan yang sederhana tanpa melibatkan banyak kode pemrograman.

Integritas dengan bahasa-bahasa pemrograman yang lain yang memungkinkan dapat digunakan oleh banyak *programmer* dengan masing-masing keahlian.

2. Fasilitas impor dari laporan pendukung format-format populer seperti *Microsoft Word*, *Excel*, *Adobe Acrobat Reader*, *HTML*, dan sebagainya.

2.9.1 Sejarah

Produk ini awalnya diciptakan oleh *Crystal Service Inc* sebagai *Quik* laporan ketika mereka tidak bisa menemukan yang cocok penulisan laporan komersial untuk perangkat akutansi. Setelah memproduksi versi 1.0 melalui 3.0 perusahaan pada tahun 1994 diakuisisi oleh *Seagate Technology*. *Crystal Report* ini digabungkan dengan sistem *holostic* untuk membentuk Kelompok Manajemen Informasi *Seagate Software*, yang kemudian namanya *Crystal Report*, dan menghasilkan versi 4.0 melalui 9.0. *Crystal Report* diakuisisi pada bulan Desember 2003 oleh *Business Object*, yang sejauh ini telah diproduksi versi 10,11(XI) dan versi 12 (2008). *Business Object* diakuisisi oleh SAP pada tanggal 8 Oktober 2007.

2.10 Database

Istilah data adalah istilah majemuk dari kata *datum*, yang berarti fakta atau bagian dari fakta yang mengandung arti yang menghubungkan dengan kenyataan, gambaran-gambaran, kata-kata, angka-angka, huruf-huruf, atau simbol-simbol, yang menunjukkan suatu ide atau objek, kondisi atau situasi dan lain-lain. Kegunaan dari data itu sendiri adalah sebagai bagian dasar dari objektif dalam proses penyusunan kebijakan dan keputusan oleh seorang pemimpin organisasi.

Pengertian data menurut *American National Standard Institute (ANSI)* :

1. Bentuk representasi dari kenyataan, konsep atau perintah dalam suatu tindakan yang dirumuskan untuk komunikasi, penafsiran, atau pengolahan.
2. Representasi sembarang sebagai karakter atau kualitas analog kemana makna itu ada atau digunakan. Secara umum diselenggarakan operasi atas data atau butir-butir data untuk membekali beberapa informasi tentang suatu entitas.

Data merupakan sumber daya yang sangat vital dalam suatu organisasi yang harus dikelola dengan baik. Karena dari data-data itulah seorang pemimpin organisasi dapat mengambil keputusan dan mengevaluasi tujuan organisasi.

Database, atau sering pula dieja *basis data*, adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil *query* basis data disebut sistem manajemen basis data (*database management system*). Sistem basis data yang dipelajari dalam ilmu informasi.

Beberapa hal yang perlu diketahui sebelum memulai perancangan

Database :

2.10.1 Data

Data masih merupakan fakta yang belum diolah. Misalnya, produksi bulan lalu berkisar 5.000 unit komputer, fakta tersebut masih berupa data.

2.10.2 Manajemen Basis Data

Di dalam manajemen basis data akan dibahas beberapa hal, yaitu :

1. Informasi

Hasil dari pengolahan data sehingga menjadi sesuatu yang berarti. Misalnya, grafik produksi komputer perbulan mengalami kenaikan sebanyak 50%, hal itu merupakan data yang sudah diolah menjadi sebuah informasi.

2. Tabel

Tabel merupakan hal yang paling mendasar dalam hal penyimpanan data pada *relational database* tabel terdiri dari *records* dan *fields* tentang sebuah topik atau katagori.

3. *Field*

Merupakan elemen dari tabel yang berisikan informasi tertentu yang spesifik seperti nama seseorang atau nama kota.

4. *Record*

Record atau disebut juga basis merupakan sekumpulan data yang berkaitan dengan sebuah subjek tertentu seperti orang, tempat, *event*, atau subjek lainnya. Setiap *record* (baris) merupakan kumpulan dari *fields* (kolom).

5. *Null*

Null berarti tidak memiliki nilai sama sekali, untuk diingat dalam terminologi komputer angka nol (0) berarti masih memiliki nilai.

6. *Relationship*

Hubungan yang dibuat antara tabel-tabel yang memiliki persamaan data dari dua *field* diantara tabel tersebut. Sebuah *Relationship* dapat berbentuk:

a. *One-to-One*

Relasi antara dua tabel yang mana kolom *primary key* dari setiap *record* dari tabel utama memiliki nilai yang sama dengan kolom dari tabel lain dan nilai kolom yang sama tersebut hanya boleh terdapat pada satu *record* saja tidak boleh lebih dari satu *record*. Sebagai ilustrasi misalnya, sebuah sepeda motor hanya dapat memiliki sebuah nomor *chasis* begitu pula sebaliknya, sebuah nomor *chasis* hanya dapat dimiliki oleh sebuah sepeda motor.

b. *One-to-Many*

Relasi antara dua buah tabel yang mana nilai *Primary Key* dari setiap *record* pada tabel utama memiliki nilai yang sama dengan banyak kolom pada tabel lain. Misalnya, Bapak A memiliki lima buah rumah, sebaliknya lima buah rumah tersebut hanya dimiliki oleh satu orang, Bapak A.

c. *Many-to-Many*

Dengan relasi *Many-to-Many* dapat dihubungkan setiap *record* dari sebuah tabel dengan banyak *record* dari tabel lain. Sebagai ilustrasi, tabel penulis dengan tabel buku, hal ini dimaksudkan agar setiap penulis dapat direlasikan dengan buku-bukunya begitu pula sebaliknya. Relasi *many-to-many* pada database dapat terjadi hanya dengan bantuan tabel penghubung yang bersisi *Primary Key* dari kedua tabel yang akan dihubungkan.

2.10.3 Teknik Merancang Basis Data




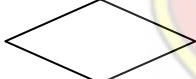
Dalam perancangan basis data dilakukan dengan dua teknik, yaitu:

1. Entitas *Relationship* Diagram.
2. Normalisasi.

1. Teknik *Entity Relationship Diagram*

ERD digunakan untuk memodelkan struktur data dan hubungan antara data karena hal ini relatif kompleks. Dengan *ERD* kita dapat menguji model dengan mengabaikan proses yang harus dilakukan. *ERD* menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data. Setiap *ERD* bisa terdapat lebih dari satu atribut. Isi atribut mempunyai sesuatu yang dapat mengidentifikasi isi *entity* satu dengan yang lain. Atribut mewakili oleh simbol *ellips* sebagai salah satu cara menggambarkan atribut.

Tabel 2.4 Simbol *ERD*

Simbol	Keterangan
	Menyatakan himpunan entitas.
	Sebagai penghubung antara himpunan relasi.
	Menyatakan atribut yang berfungsi sebagai <i>Key</i> .
	Menyatakan himpunan relasi.
1 dan 1, 1 dan N, N dan N	Relasi dapat dinyatakan dengan pemakaian angka (1 & 1 untuk relasi satu-ke-satu, 1 & N untuk relasi satu-ke-banyak atau N & N untuk relasi banyak-ke-banyak).

2. Teknik Normalisasi

Normalisasi adalah suatu proses untuk membuat data yang tidak normal menjadi data yang normal, yang dapat menyebabkan *anomaly* pada saat terjadi operasi manipulasi data seperti : *insert*, *edit*, *delete*.

Sebelum normalisasi dilakukan harus diketahui bahwa setiap tabel di dalam *database* mempunyai nama tabel yang unik yang mengidentifikasi isinya.

Sebuah relasi adalah istilah resmi untuk tabel. Sebuah baris atau *record* pada tabel disebut *Tuple*. Tiap kolom pada tabel mempunyai nama kolom yang berbeda. Banyak *Tuple* pada sebuah relasi disebut *Cardinality* dan banyak atribut/kolom disebut *Degree*.

Dalam suatu relasi terdapat atribut-atribut yang membedakan *entity* yang satu dengan *entity* yang lainnya. Di dalam suatu atribut terdapat kunci yang dapat membedakan suatu *field* dengan *field* berikutnya. *Key* atau kunci adalah suatu properti yang menentukan apakah suatu kolom pada tabel sangat penting atau tidak. Berikut bentuk-bentuk *Key*, yaitu :

1. *Candidate Key* adalah sebuah Atribut atau lebih secara unik mengidentifikasi sebuah baris, Atribut ini mempunyai nilai yang unik pada setiap barisnya, yang berfungsi sebagai calon *Primary Key*.
2. *Primary Key* adalah *Candidate Key* yang anda pilih untuk mengidentifikasi tiap baris secara unik, yang harus merupakan *field* yang benar-benar unik dan tidak boleh ada nilai *Null* yang mempunyai duplikat.
3. *Alternate Key* adalah *Candidate Key* yang tidak dipilih sebagai *Primary Key*.
4. *Composite Key* adalah penambahan kolom lain sebagai *Primary Key* dikarenakan satu atribut tidak dapat dipilih untuk mengidentifikasi baris secara unik.
5. *Foreign Key* adalah sebuah *Primary Key* yang terhubung ke tabel lain yang berfungsi sebagai penghubung.

Sebuah tabel dikatakan sudah normal apabila :

- a. Nilai tiap kolom hanya satu.
- b. Semua nilai pada setiap kolom harus sesuai dengan tipe data kolom

tersebut.

- c. Nama kolom harus unik dimana setiap kolom memiliki nama yang berbeda.
- d. Tidak ada 2 *record* yang sama.
- e. Urutan kolom tidak signifikan atau tidak penting.
- f. Urutan *record* tidak juga signifikan.

Berikut bentuk level-level normalisasi yang digunakan dalam penyelesaian masalah yang ada dalam penelitian ini, yaitu sebagai berikut :

1. *First Normal Form (1NF)*

Suatu tabel yang dikatakan dalam keadaan *First Normal Form (1NF)* jika:

- a. Tidak ada perulangan *record* data dalam tabel.
- b. Setiap sel memiliki satu nilai saja. Artinya tidak ada perulangan group atau *array*.
- c. Data yang diinputkan memiliki tipe data yang sama dengan tipe data kolom dalam tabel.

2. *Second Normal Form (2NF)*

Suatu tabel dikatakan dalam keadaan *Second Normal Form (2NF)* jika tabel tersebut sudah dalam keadaan *First Normal Form (1NF)* dan jika semua atribut yang bukan kunci tabel, baik *primary key* maupun *foreign key* tergantung pada semua kunci dalam tabel.

3. *Third Normal Form (3NF)*

Suatu tabel dikatakan dalam keadaan *Third Normal Form (3NF)* jika tabel tersebut sudah dalam keadaan *Second Normal Form (2NF)* dan jika tidak terdapat ketergantungan yang transitif. Artinya data-data yang mungkin diisi berulang-ulang dapat dibuat sebuah tabel baru.