

SISTEM PENJUALAN TUNAI PADA PT. DJOE I SOE MENGUNAKAN DELPHI DENGAN PERANCANGAN SISTEM BERORIENTASI OBJEK

Ina Agustina dan Tri Wahyu Widyaningsih

Jurusan Sistem Informasi Universitas Nasional
Jl. Sawo Manila No. 61, Pasar Minggu Jakarta Selatan 12520
Telp. 7891753, Fax. 7891753
E-Mail : ina_agustina2007@yahoo.com;widya_ayub@yahoo.co.id

ABSTRAK

Pada Penelitian ini kami membahas tentang pemanfaatan Borland Delphi sebagai salah satu pemrograman berorientasi objek untuk memperoleh rancangan interface yang lebih menarik, dan mendukung pemrograman database dalam membuat suatu sistem penjualan tunai yang ada pada PT. DJOE I SOE sehingga memudahkan user dalam pencarian data.

Pembahasan diawali dengan memperkenalkan sistem penjualan dan Borland Delphi. Pada sistem penjualan, kami akan membahas tentang pembuatan UML dan flowchart. Pada Borland Delphi kami akan membahas tentang perintah-perintah yang ada pada Borland Delphi beserta fungsinya.

Kami membuat suatu rancangan sistem penjualan tunai barang secara terkomputerisasi dengan efisien dan praktis seperti di bawah ini yaitu Pertama Proses pengisian data menjadi lebih mudah karena dapat dilakukan secara otomatis sehingga dapat mengurangi tingkat kesalahan. Yang Kedua ialah Proses pencetakan laporan penjualan tunai barang dapat dibuat berdasarkan data-data persediaan sehingga proses pencetakan menjadi lebih mudah dan cepat.

Kata Kunci : Perancangan Sistem Berorientasi Objek, Delphi, UML

1. PENDAHULUAN

Pada sistem penjualan perusahaan ini ternyata mempunyai banyak kelemahan, antara lain tingginya tingkat kesalahan manusia (*human error*) sehingga sering terjadi adanya salah perhitungan pada pembuatan laporan penjualan, dimana laporan ini dibuat oleh bagian *accounting*. Begitupula pada proses pencatatan akuntansi yang dilakukan secara manual, yang membutuhkan waktu yang cukup lama dalam pengerjaannya. Permasalahan di atas menyebabkan pendapatan dan kinerja perusahaan yang relatif menurun, yang menyebabkan jumlah pesanan yang tidak

bisa ditangani dengan baik. Itu semua dikarenakan karena sistem yang dikerjakan masih banyak kekurangannya. Untuk itu dibuat sebuah sistem baru.

2. TINJAUAN PUSTAKA

2.1 Pengertian Sistem

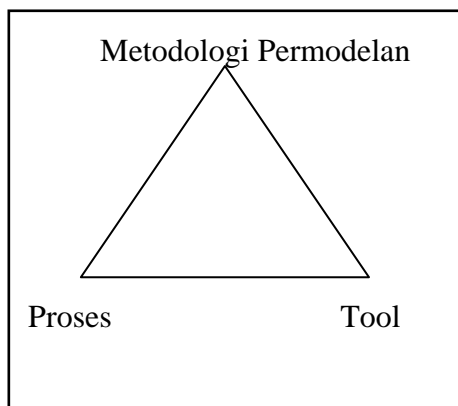
Pengertian sistem yang menekankan pada prosedurnya yaitu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.

Pengertian sistem yang menekankan pada elemen atau komponennya yaitu sistem

adalah kumpulan dari elemen-elemen yang berinteraksi atau berhubungan satu sama lainnya dan bertanggung jawab memproses masukan (input) sehingga menghasilkan keluaran (output).

2.2 Unified Modelling Language (UML)

UML (Unified Modelling Language) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Kesuksesan suatu permodelan piranti lunak ditentukan oleh tiga unsure, yang kemudian terkenal dengan sebuah segitiga sukses (*the triagle of success*). Ketiga unsur tersebut adalah metode permodelan (*notation*), proses (*process*) dan *tool* yang digunakan.



Gambar 1. Notasi Permodelan

UML merupakan kesatuan dari bahasa permodelan yang dikembangkan oleh Booch, *Object Medelling Technique* (OMT), *Object Oriented Software Engineering* (OOSE). Metode Booch dari Grady Booch terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan design ke dalam empat tahapan iterative, yaitu : *identifikasi kelas-kelas* dan *obyek-obyek*, *identifikasi semantik* dari hubungan obyek dan kelas tersebut, *perincian interface* dan *implementasi*. Keunggulan metode Booch

adalah pada detilnya notasi dan elemen. Permodelan OMT yang dikembangkan oleh Rumbaugh didasarkan pada *analisis terstruktur* dan *pemodelan entity-relationship*. Tahapan utama dalam metodologi ini adalah analisis, design sistem, design obyek dan implementasi. Keunggulan metode ini adalah dalam penotasian yang mendukung semua konsep OO. Metode OOSE dari Jacobson lebih memberi penekanan pada use case. OOSE memiliki tiga tahapan yaitu membuat *model requirement dan analisis*, *design dan implementasi*, dan *model pengujian* (test model). Keunggulan model ini adalah mudah dipelajari karena memiliki notasi yang sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak.

Abstraksi konsep dasar UML yang terdiri dari *structural classification*, *dynamic behavior*, dan *model management Diagrams*. Main concepts bisa kita pandang sebagai term yang akan muncul pada saat kita membuat diagram. Dan view adalah kategori dari diagram tersebut.

Untuk menguasai UML, sebenarnya cukup dua hal yang harus kita perhatikan:

1. Menguasai pembuatan diagram UML
2. Menguasai langkah-langkah dalam analisa dan pengembangan dengan UML.

2.2.1 Use case diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah system. Yang ditekankan adalah “apa” yang diperbuat system, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan system. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke system, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah

actor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah system, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua *feature* yang ada pada system.

Sebuah *use case* dapat meng-include fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di include akan dipanggil setiap kali *use case* yang meng-include dieksekusi secara normal. Sebuah *use case* dapat di-include oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas common.

Sebuah *use case* juga dapat meng-extend *use case* lain dengan behaviour-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

2.2.2 Class Diagram

Class adalah sebuah spesifikasi yang jika di instansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/property) suatu system, sekaligus menawarkan layanan untuk manipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi class, package dan objek serta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama(stereotype)
2. Atribut

3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

- a. *Private*, tidak dapat dipanggil dari luar class yang bersangkutan
- b. *Protected*, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
- c. *Public*, dapat dipanggil oleh siapa saja

Class dapat merupakan implementasi dari sebuah interface, yaitu class abstrak yang hanya memiliki metoda. Interface tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah class. Dengan demikian interface mendukung resolusi metoda pada saat run-time. Hubungan antar *Class* :

1. Asosiasi, yaitu hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui eksistensi class lain. Panah navigability menunjukkan arah query antar class.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”)
3. Pewarisan, yaitu hubungan yang menyatakan hirarkis antar class. *Class* dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda class asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari class yang diwarisinya. Kebaikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (message) yang di-passing dari satu class kepada class lain. Hubungan dinamis dapat digambarkan dengan menggunakan sequence diagram.

2.3 Borland Delphi 7.0

Borland Delphi yang selanjutnya disingkat dengan Delphi merupakan

program aplikasi database yang berbasis Object Pascal, yang juga memberikan fasilitas pembuatan aplikasi visual seperti Visual Basic..

Memakai Delphi adalah cara yang paling sederhana untuk membangun aplikasi berbasis windows. Bagi sebagian orang, jawaban tersebut kurang memuaskan. Oleh sebab itu, diperlukan jawaban panjang. Produktifitas dari pengembangan perangkat lunak dibagi menjadi 5 atribut penting, yaitu :

- a. *Kualitas* lingkungan pengembangan visual
- b. *Kecepatan* kompilasi dibandingkan dengan kompleksitasnya
- c. *Kekuatan* bahasa pemrograman dibandingkan dengan kekompleksitasnya
- d. *Fleksibilitas* arsitektur basis data
- e. *Pola desain dan pemakaian* yang diwujudkan oleh frameworknya

2.4 Pengertian Database

Suatu database didefinisikan sebagai kumpulan data yang disatukan dalam suatu organisasi. Suatu organisasi itu dapat berupa perusahaan, bank, universitas, dan bidang usaha lain yang berhubungan dengan database.

Tujuan dari pembuatan database biasanya untuk mempermudah suatu organisasi yang dalam hal mengolah data, karena data adalah informasi yang sangat penting untuk jangka waktu yang panjang dalam suatu organisasi.

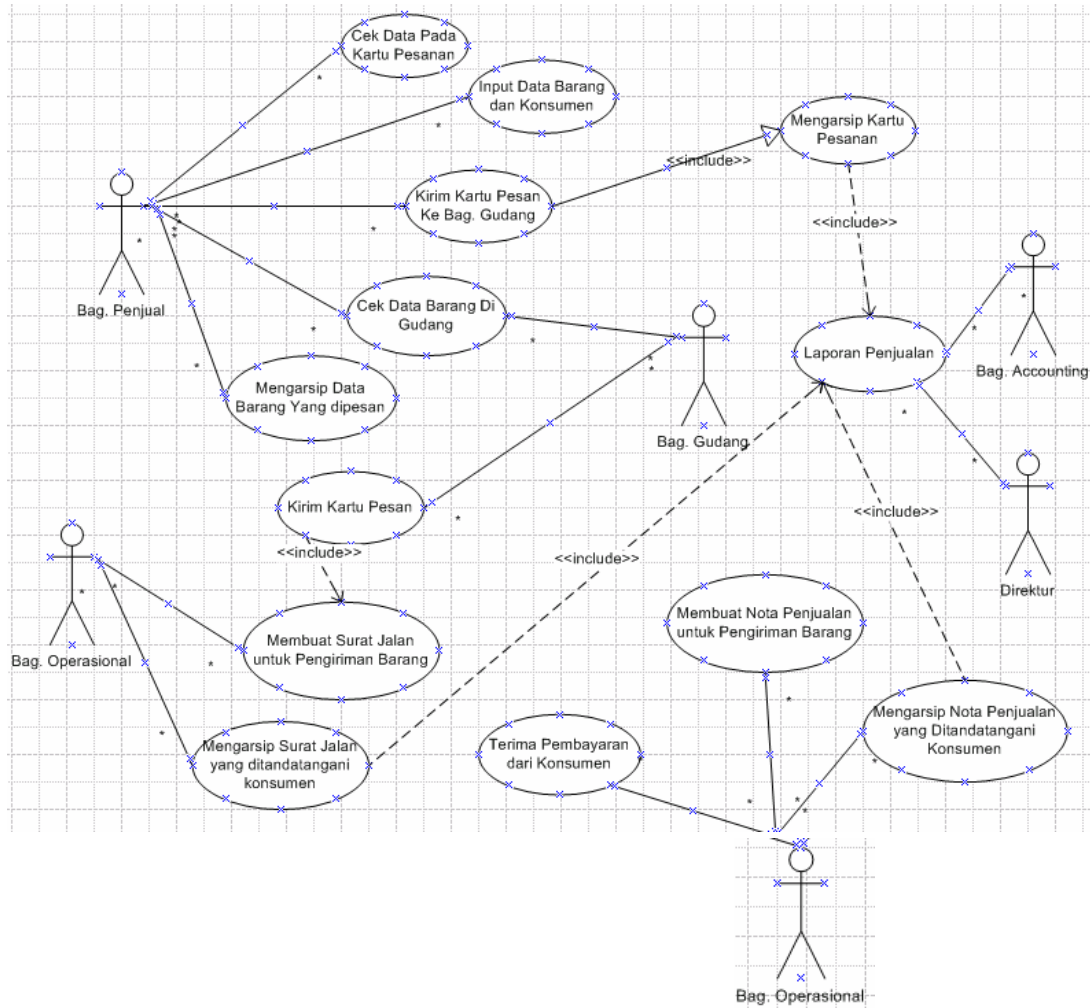
Mengolah data dalam hal ini adalah database mampu menyimpan data, mengupdate data, menambahkan data, menghapus data, dll. Makin pentingnya sebuah database, maka makin banyak pula software – software yang menunjang database. Software – software itu antara lain : Oracle, DB2, MySQL, MS Access, Informix, Interbase, dll.

Pada pembuatan sistem ini penulis menggunakan software pemrograman Borland Delphi 7.0 yang berorientasi object, dan sebagai databasenya penulis menggunakan database standard Borland Delphi 7.0 yaitu Paradox 7.0. Untuk menghubungkan antara program dan databasenya maka dibutuhkan sebuah Alias.

3. METODE PENELITIAN

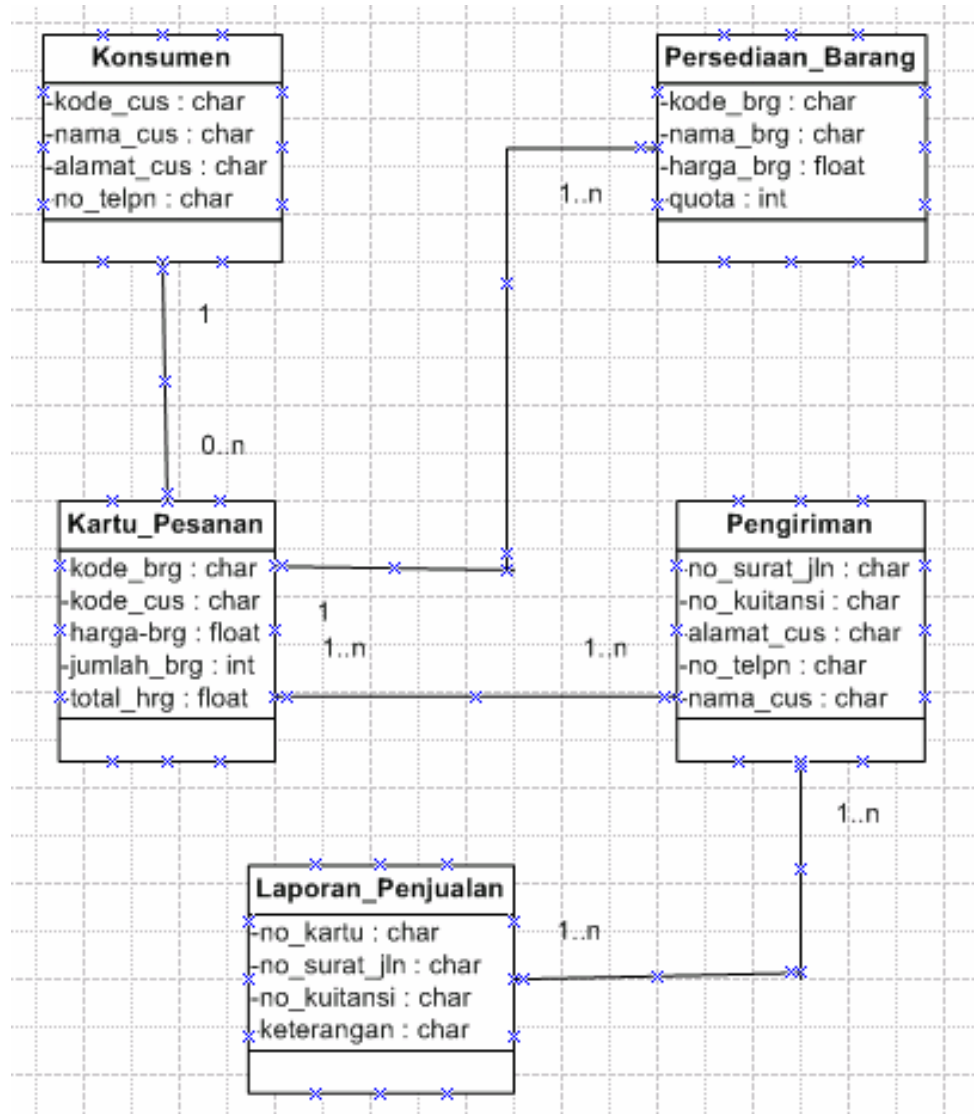
Dalam penyusunan penelitian ini, untuk memperoleh informasi data pengumpulan data, digunakan tiga metode sebagai berikut: Metode Kepustakaan (*Library*), dengan membaca *literature* yang berhubungan dengan penelitian ini; Metode Observasi (*Observation*), dengan cara meninjau langsung kepada bagian sistem penjualan pada PT. DJOE 1 SOE; Metode Wawancara (*Interview*), dengan cara mengadakan komunikasi langsung atau tatap muka dengan bagian penjualan PT. DJOE 1 SOE

4. PEMBAHASAN



Gambar 3. Gambar Diagram Use Case

Diagram Class



Gambar 4. Gambar Diagram Class

4. PENUTUP

Kesimpulan

Untuk memecahkan permasalahan yang ada, maka diusulkan suatu rancangan sistem penjualan tunai barang secara terkomputerisasi dengan efisien dan praktis seperti di bawah ini :

- a. Proses pengisian data menjadi lebih mudah karena dapat dilakukan secara otomatis sehingga dapat mengurangi tingkat kesalahan.
- b. Proses pencetakan laporan penjualan tunai barang dapat dibuat berdasarkan data-data persediaan sehingga proses pencetakan menjadi lebih mudah dan cepat

Saran

Penulis memberikan saran yang kiranya dapat berguna bagi PT.DJOE I SOE untuk meningkatkan pengolahan data keuangan yaitu :

- a. Diadakan upaya untuk menjadi personal yang terkait langsung dalam penggunaan teknologi informasi komputer agar pengolahan data dapat dilakukan dengan benar.
- b. Perlu adanya backup data agar data ganda agar dapat tersimpan dengan aman.
- c. Pemeliharaan hardware perlu disusun jadwal berkala misalnya dengan melakukan service dengan perusahaan komputer yang bertanggungjawab dalam masalah hardware.
- d. Perlu dibentuk bantuan khusus yang terdiri dari orang-orang yang bertanggungjawab dalam mengembangkan tugas yang ada.
- e. Perlu adanya backup data yang mengantisipasi kerusakan data yang terkena virus.

6. DAFTAR PUSTAKA

- Abdul Kadir, *Dasar Pemrograman Delphi 5.0 Jilid 1*, Salemba Infotek, Jakarta, 2001
- Fathansyah, Jr. (2001), *Basis Data*, Bandung. Penerbit Informatika.
- Inge Martina, *Database Client/Server Menggunakan Delphi*, Elex Media Komputindo,
- Jogiyanto, HM. (1999), *Analisis & Disain Sistem Informasi : Pendekatan perstruktur Teori dan praktek Aplikasi Bisnis*, Yogyakarta: Andi.
- Jayanto, *Membuat Aplikasi Database dengan Delphi*, Elex Media Komputindo, Jakarta, 1999
- Kadir, Abdul (1999), *Konsep dan Tuntunan I'rukn.s Basis Data*, Yogyakarta: Andi.
- Kristanto, Harianto, Jr. (1994), *Konsep dan Perancangan Database*, Yogyakarta: Andi Offset.
- Kurniawan, Dios (1997), *HTML 3 untuk Publikasi di Internet*, Yogyakarta: BPFE
- M. Agus J Alam, *Belajar Sendiri Borland Delphi 6.0*, Media Komputindo, Jakarta, 2001
- M McLeod, Raymond (1996), *Sistem Informasi Manajemen*, Edisi Bahasa Indonesia Jilid I, Jakarta: PT. Prenhallindo.
- Munawar, *Pemodelan Visual dengan UML*, Graha Ilmu, Yogyakarta, 2005
- Silberschatz, Abraham, dkk (2002), *Database Sistem System Concepts*, Edisi Keempat, McGraw-Hill.Jakarta, 2002
- Suryadi H S, *Sistem Basis Data*, Gunadarma, 1996