

RANCANG BANGUN APLIKASI BARANG JASA MENGUNAKAN C#, WPF, DAN SQL SERVER 2012 (Studi Kasus PT. X)

Reza Kamaluddin Isman¹, Desi Dyah Sulistyarini², Prihatin Oktivasari³

Teknik Informatika dan Komputer Politeknik Negeri Jakarta

Jl. Prof. G.A. Siwabessy, Kampus Baru UI, Kota Depok Jawa Barat 16424

¹desidyahs@gmail.com

²rezakisman@gmail.com

³ti2n.oktivasari@yahoo.com

Abstract

Barang Jasa is an application that used for Divisi Pengadaan Barang dan Jasa. Barang Jasa serves to record every purchase of goods and services used to meet the needs of both fleet and crew in the long term. Barang Jasa also serves to make the Nota Dinas to be processed by Divisi Keuangan on the purchase of goods and services that have been performed. This application was created by using Windows Presentation Foundation (WPF), which is a Microsoft technology to create desktop applications. This application was made using the method prototyping development, C# as a programming language, WPF as a interface design technology, and SQL Server 2012 as the database. WPF can present the appearance that can be tailored to user needs. This study discusses the interface, backend applications, and database design. After testing the flexibility of the interface on the screen with a resolution of 1366 x 768, 1024 x 768 and 1280 x 720, proves that WPF can adjust the screen resolution being tested. The test results showed that the backend of Barang Jasa application are functioning properly, the test is done using sql query by entering the required parameters, for example username and password to log in and match the query returns results with what is displayed by the application.

Keywords— C#, prototyping, SQL Server 2012, WPF.

PENDAHULUAN

Divisi Pegadaan Barang dan Jasa pada PT. X mengalami kesulitan dalam menginput data barang maupun jasa kebutuhan perusahaan. Aplikasi Barang Jasa dibuat untuk memudahkan Divisi Pengadaan Barang dan Jasa dalam menginput data barang dan jasa yang digunakan untuk menyuplai kebutuhan armada maupun anak buah kapal dalam jangka panjang. Selain itu, aplikasi ini juga memudahkan pembuatan Nota Dinas terhadap tagihan barang dan jasa yang dibeli untuk kebutuhan armada untuk selanjutnya dapat diproses oleh Divisi Keuangan. Selanjutnya aplikasi ini juga memiliki laporan terhadap pembelian barang dan jasa yang telah dilakukan oleh Divisi Pengadaan Barang dan Jasa.

Dalam pembuatan aplikasi Barang Jasa, WPF digunakan untuk membuat antar muka aplikasi, C# digunakan untuk membuat back end aplikasi, SQL Server 2012 digunakan untuk membuat database yang akan digunakan pada Barang, dan penggunaan

kerangka kerja .NET Framework 4.5.2 yang dibuat oleh Microsoft.

TINJAUAN PUSTAKA

Visual Studio 2015

Microsoft [1] menyatakan bahwa *Microsoft Visual Studio 2015* merupakan alat yang digunakan untuk membuat sebuah perangkat lunak, dimulai dengan desain *User Interface* atau tahap perencanaan, pengkodean, pengujian, *debugging*, analisis kualitas kode dan kinerja, membagikan ke pelanggan (*publishing*), dan mengumpulkan telemetri (*metered variable*) pada penggunaan. Alat-alat ini dirancang untuk bekerja secara bersama-sama dengan semulus mungkin, dan semua dapat dikerjakan melalui *Visual Studio Integrated Development Environment (IDE)*.

C#

Margaret Rouse [2] berpendapat bahwa C# adalah bahasa pemrograman berbasis objek yang dibuat *Microsoft* yang mengombinasikan kekuatan bahasa pemrograman C++ dan bahasa pemrograman *Visual*

Basic. C# merupakan bahasa pemrograman yang berbasis C++ dan mengandung fitur-fitur yang serupa dengan bahasa pemrograman Java.

Selain itu Margaret Rouse [3] juga menyatakan bahwa C# merupakan bahasa pemrograman yang mendukung *object-oriented programming* (OOP). OOP sendiri merupakan metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada. Jadi setiap bagian dari suatu permasalahan adalah objek, objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi.

.NET Framework

.NET Framework merupakan kerangka kerja yang digunakan para pengembang aplikasi yang berbasis Windows. .NET menyediakan beragam pilihan untuk digunakan dalam pembuatan aplikasi mulai dari aplikasi *desktop* (Windows Forms), *web-accessible application* (ASP.NET), hingga *System.XML* [4]. .NET adalah platform pengembangan yang digunakan oleh jutaan pengembang untuk membuat aplikasi dan layanan pada berbagai perangkat dan OS, dengan performa luar biasa dan produktivitas pengembang yang bagus. .NET sangat cepat. Perangkat aplikasi untuk Windows, iOS, dan Android berjalan secara *native* sehingga pengembang dapat membuat aplikasi dengan maksimal yang dapat menggunakan semua kemampuan perangkat.

SQL Server 2012

Stan Giblisco [5] berpendapat bahwa SQL Server 2012 adalah sistem manajemen database relasional (RDBMS) dari Microsoft yang dirancang untuk lingkungan perusahaan. SQL Server berjalan pada T-SQL (*Transact -SQL*), satu set ekstensi pemrograman dari Sybase dan Microsoft ditambah beberapa fitur untuk SQL standar, termasuk transaksi yang dikendalikan, *exception* dan *error handling*, pengolahan baris, dan *declared variables*.

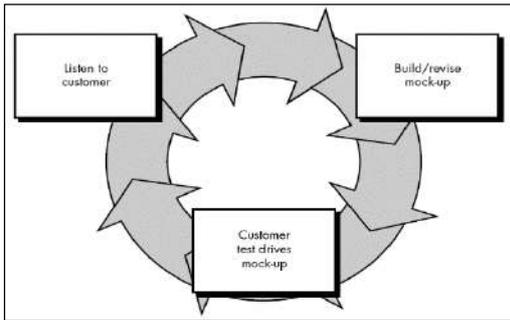
Windows Presentation Foundation (WPF)

Dalam situsnya Microsoft [6] menulis bahwa *Windows Presentation Foundation* (WPF) memungkinkan dalam pembuatan aplikasi *desktop client* untuk windows dengan *visual user experiences* yang menakjubkan. Inti dari WPF adalah sebuah mesin render yang beresolusi-independen dan berbasis vektor yang dibangun untuk mengambil

keuntungan dari hardware grafis modern. WPF memperluas intinya dengan seperangkat fitur pengembangan aplikasi yang mencakup *Extensible Application Markup Language* (XAML), kontrol, *data binding*, tata letak, grafis 2-D dan 3-D, animasi, gaya, *template*, dokumen, media, teks, dan tipografi. WPF termasuk dalam .NET Framework, sehingga dapat dibangun aplikasi yang menggabungkan unsur-unsur lain dari .NET Framework *class library*. Microsoft [7] juga menyatakan bahwa XAML adalah bahasa markup deklaratif. Seperti diterapkan pada model pemrograman .NET Framework, XAML menyederhanakan pembuatan UI untuk aplikasi .NET Framework. Elemen UI bisa dibuat agar terlihat di markup deklaratif XAML, dan kemudian memisahkan definisi UI dari logika run-time dengan menggunakan *file code-behind*, tergabung ke *markup* melalui definisi kelas parsial. XAML langsung mewakili instansiasi objek dalam satu set spesifik jenis dukungan didefinisikan dalam majelis. Tidak seperti kebanyakan bahasa markup lainnya, yang biasanya ditafsirkan langsung ke sistem. XAML memungkinkan alur kerja di mana pihak yang terpisah dapat bekerja pada UI dan logika aplikasi, menggunakan alat berpotensi berbeda. Ketika direpresentasikan sebagai teks, file XAML adalah file XML yang umumnya memiliki ekstensi *.xaml*.

METODE PENELITIAN

Prototyping digunakan sebagai metode pengembangan aplikasi yang digunakan untuk membuat aplikasi Barang Jasa. Wawancara kepada user yang berkepentingan digunakan untuk mendapatkan user requirements. Setelah user requirements didapat, sebuah initial prototype disajikan, sehingga user dapat memahami bagaimana bentuk dan cara kerja aplikasi yang dibuat. Setelah user melihat tampilan dan juga cara kerja dari aplikasi yang dibuat, didapatkan saran dan juga masukan mengenai aplikasi Barang Jasa. Saran dan masukan dari user digunakan untuk memperbaiki aplikasi, sehingga aplikasi dapat disesuaikan dengan keinginan dari user dan kebutuhan dari user.



Gambar 1. Prototyping model

HASIL DAN PEMBAHASAN

Dalam pembuatan aplikasi Barang Jasa, hal yang dilakukan pada tahap analisis kebutuhan adalah dengan melakukan interview kepada user aplikasi. Setelah interview dilakukan, kemudian *requirements* didapatkan. Beberapa *requirements* antara lain:

- Menggunakan *database* yang terintegrasi.
- Tambah barang dan jasa.
- Tambah kapal.
- Buat invoice dan lihat invoice.
- Laporan barang dan jasa.
- Lihat detail barang dan jasa.

Setelah mendapat *requirements* tersebut, pembuatan aplikasi dilakukan. Pembuatan aplikasi Barang Jasa dikelompokkan menjadi tiga tahapan, yaitu perancangan data menggunakan SQL Server 2012, pembuatan antar muka (interface) menggunakan WPF, dan pembuatan back end aplikasi menggunakan bahasa C#. WPF dan C# terdapat pada kerangka kerja .NET Framework yang terdapat pada Visual Studio 2015.

Perancangan Data

Proses perancangan data dimulai dengan mengumpulkan data *requirements* dari user. Pada kasus ini, PT. X memberikan file excel (gambar 2) yang berisi laporan suplai barang. Kemudian file excel tersebut dijadikan acuan untuk membuat database aplikasi. Setelah dibuat prototype perancangan data yang dikonsultasikan kepada user berkali kali, kemudian dibuat table-table pada database menggunakan query sql (gambar 3).

NO	NAMA BARANG	JENIS	SATUAN	Jumlah	HARGA	TOTAL	KETERANGAN
1	LF Lub Oil ME 670			12.00	Rp 171,500	Rp 2,058,000	21 Januari Tarahan
2	LF Lub Oil ME 777			8.00	Rp 255,000	Rp 2,040,000	21 Januari Tarahan
3	FF FO ME 202			6.00	Rp 250,000	Rp 1,500,000	21 Januari Tarahan
4	FS Racor ME 19763			12.00	Rp 270,000	Rp 3,240,000	21 Januari Tarahan
5	AF Air Cleaner ME 4553 M			6.00	Rp 1,900,000	Rp 11,400,000	21 Januari Tarahan
6	LF Lub Oil AE 3345			5.00	Rp 565,000	Rp 2,825,000	21 Januari Tarahan
7	FS FO AE 1280			5.00	Rp 150,000	Rp 750,000	21 Januari Tarahan
8	FS FO AE 5327			5.00	Rp 185,000	Rp 925,000	21 Januari Tarahan
9	Racor AE 2010			6.00	Rp 120,000	Rp 720,000	21 Januari Tarahan
10	Air Filter Element 5-14215038-0			4.00	Rp 225,000	Rp 900,000	21 Januari Tarahan
11	Water Coolant PF Plus			20.00	Rp 235,000	Rp 4,700,000	21 Januari Tarahan
12	LO Meditran SX			2.00	Rp 5,800,000	Rp 11,600,000	21 Januari Tarahan
13	Mesin Gerinda potong			1.00	Rp 782,480	Rp 782,480	21 Januari Tarahan
14	Radio Handy Talky IC-M34			2.00	Rp 2,500,000	Rp 5,000,000	21 Januari Tarahan
15	Lampu Tongkang Signalite			2.00	Rp 395,000	Rp 790,000	21 Januari Tarahan
16	Ongkos Kirim					Rp 5,000,000	21 Januari Tarahan
TOTAL						54,230,480.00	

Gambar 2. File excel laporan supply barang

```

CREATE DATABASE BAG
GO
CREATE SCHEMA Transactions
GO
CREATE TABLE Transactions.Header
(
    HeaderNumber VARCHAR (10) PRIMARY
    KEY,
    HeaderTotal MONEY,
    HeaderDate DATE,
    EmployeeNIP VARCHAR (10),
    ShipID INT
)
GO
CREATE TABLE Transactions.Detail
(
    DetailItemName NTEXT,
    DetailItemType NTEXT,
    DetailItemUnit NTEXT,
    DetailItemQuantity DECIMAL(18,0),
    DetailItemPrice MONEY,
    DetailItemTotalPrice MONEY,
    DetailItemSupplier VARCHAR(100),
    DetailItemSupplyDate DATE,
    DetailItemSupplyPlace NTEXT,
    HeaderNumber VARCHAR (10)
)
GO
CREATE TABLE Transactions.Invoice
(
    InvoiceNomor VARCHAR(16) PRIMARY
    KEY,
    InvoicePerihal NTEXT,
    InvoiceJumlah NTEXT,
    InvoiceTanggalDiajukan DATE,
    InvoiceTanggalSupply DATE,
    InvoiceSupplier VARCHAR(100),
    InvoiceTotal MONEY,
    InvoiceTerbilang NTEXT,
    InvoiceManager NTEXT,
    EmployeeNIP VARCHAR(10)
  
```

Gambar 3. Query SQL table transaction header dan detail

Pembuatan Interface

Setelah dibuat table pada SQL Server 2012, kemudian dilakukan pembuatan interface dari aplikasi Barang Jasa. Interface dibuat dengan menggunakan Windows Presentation Foundation (WPF). Window yang pertama kali dibuat adalah Login. Window_Login (gambar 4) memiliki dua text box yang harus diisi, yaitu username dan password. Window_Login juga memiliki tombol berwarna biru untuk masuk ke dalam aplikasi, tombol hijau untuk bantuan, tombol merah muda untuk pengaturan, tombol kuning digunakan jika user lupa dengan password yang dimiliki, dan tombol merah untuk keluar dari aplikasi.



Gambar 4. Window_Login aplikasi Barang Jasa

Setelah login berhasil dilakukan, maka Window_MainMenu muncul. Namun jika login gagal dilakukan, akan muncul message box jika terdapat kesalahan dalam username, password, maupun koneksi dari database. Berikut ini ditampilkan pada gambar 5 Window_MainMenu. Terdapat enam menu utama dari aplikasi Barang Jasa, yaitu Tambah Barang dan Jasa, Tambah Kapal, Invoice, Lihat Invoice, Laporan, dan Lihat Detail Barang dan Jasa. Selain itu terdapat lima menu tambahan yang digunakan untuk melengkapi aplikasi Barang Jasa, yaitu Dashboard, My Profile, Help, About, dan Logout.



Gambar 5. Window_MainMenu aplikasi Barang Jasa

Sebagai contoh jika memilih menu Tambah Barang dan Jasa, maka Window_TambahBarangJasa terlihat seperti pada gambar 6. User harus mengisi field nomor transaksi dengan memilih tombol generate nomor transaksi berwarna oranye yang telah disediakan. Selain itu, user harus mengisi tanggal transaksi dan juga kapal yang menjadi tujuan pengiriman barang. Langkah selanjutnya yang harus dilakukan oleh user adalah memilih untuk menambah barang/jasa secara manual atau menambah data berdasarkan file excel yang telah dibuat.



Gambar 6. Window_TambahBarangJasa aplikasi Barang Jasa

Jika user memilih untuk menambahkan barang/jasa secara manual, maka Window_TambahBarangJasaManual terlihat seperti pada gambar 7. User diwajibkan untuk mengisi semua field yang disediakan untuk melengkapi data-data yang dibutuhkan oleh aplikasi Barang Jasa



Gambar 7. Window_TambahBarangJasaManual aplikasi Barang Jasa

Pada gambar 8 terdapat Window_Invoice jika pada Window_MainMenu memilih menu Invoice. Menu Invoice digunakan untuk membuat invoice terhadap pengajuan jumlah tagihan yang akan ditujukan kepada Divisi Keuangan perusahaan. User harus mengisi semua field yang telah disediakan untuk dapat melanjutkan ke proses pencetakan invoice.



Gambar 8. Window_Invoice aplikasi Barang Jasa

Pada gambar 9 terdapat invoice yang sudah siap dicetak setelah user mengisi field yang sebelumnya telah disediakan.



Gambar 9. Window_CetakInvoice aplikasi Barang Jasa

Pembuatan Backend

Setelah dibuat interface aplikasi, kemudian dibuat backend aplikasi Barang Jasa. Backend aplikasi

berisi logika bisnis, fungsi, dan validasi aplikasi. Berikut merupakan backend aplikasi untuk menjalankan fungsi Add Invoice (gambar 10). Selain fungsi Add Invoice juga terdapat beberapa fungsi lain, diantaranya: Invoice DGV, Bind Invoice DGV, Auto Generate Invoice, Bind Invoice Detail, Edit Invoice, Add Transactions, Transactions Excel, dan Login. Seluruh fungsi tersebut dibuat dengan mengonsultasikan kepada user dengan metode prototyping.

```

public void AddInvoice(string InvoiceNomor,
string InvoicePerihal, string InvoiceJumlah,
string InvoiceTanggalDiajukan, string
InvoiceTanggalSupply, string InvoiceSupplier,
decimal InvoiceTotal, string InvoiceTerbilang,
string InvoiceManager, string EmployeeNIP)
{sqlcon = con.ConnectSQL(); using
(sqlcon){sqlcon.Open();
string sql = "INSERT INTO
Transactions.Invoice VALUES(@InvoiceNomor,
@InvoicePerihal, @InvoiceJumlah,
@InvoiceTanggalDiajukan, @InvoiceSupplier,
@InvoiceTotal, @InvoiceTerbilang,
@InvoiceManager, @EmployeeNIP)";
SqlCommand sqlcom = new SqlCommand(sql,
sqlcon); using
(sqlcom){ sqlcom.Parameters.Add(new
SqlParameter("@InvoiceNomor", InvoiceNomor));
sqlcom.Parameters.Add(new
SqlParameter("@InvoicePerihal",
InvoicePerihal));
sqlcom.Parameters.Add(new
SqlParameter("@InvoiceJumlah",
InvoiceJumlah));
sqlcom.Parameters.Add(new
SqlParameter("@InvoiceTanggalDiajukan",
InvoiceTanggalDiajukan));
sqlcom.Parameters.Add(new
SqlParameter("@InvoiceTanggalSupply",
InvoiceTanggalSupply));
sqlcom.Parameters.Add(new
SqlParameter("@InvoiceSupplier",
InvoiceSupplier));
sqlcom.Parameters.Add(new
SqlParameter("@InvoiceTotal", InvoiceTotal));
sqlcom.Parameters.Add(new
SqlParameter("@InvoiceTerbilang",
InvoiceTerbilang));
sqlcom.Parameters.Add(new
SqlParameter("@InvoiceManager",
InvoiceManager));
sqlcom.Parameters.Add(new
SqlParameter("@EmployeeNIP", EmployeeNIP));
sqlcom.ExecuteNonQuery();}
sqlcon.Close();}
    
```

Gambar 10. Fungsi Add Invoice

Pengujian Aplikasi

Setelah aplikasi dibuat, maka pengujian dilakukan terhadap interface dan backend aplikasi. Pengujian interface aplikasi dilakukan dengan mengukur kemampuan fleksibilitas WPF pada resolusi layar yang berbeda-beda. Dimulai dari resolusi native laptop pada umumnya, yaitu 1366 x 768, kemudian resolusi 1024 x 768, dan resolusi HD (1280 x 720). Gambar 11 merupakan hasil pengujian pada resolusi layar 1366 x 768. Kemudian pada gambar 12 merupakan hasil pengujian pada resolusi HD (1280 x 720). Terakhir pada gambar 13 merupakan hasil pengujian pada resolusi 1024 x 768.



Gambar 11. Tampilan aplikasi pada resolusi 1366 x 768

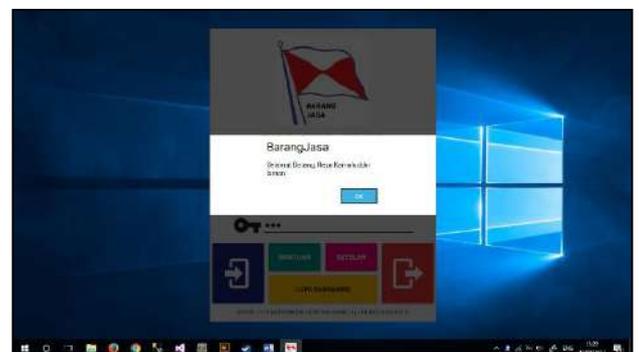


Gambar 12. Tampilan aplikasi pada resolusi HD (1280 x 720)

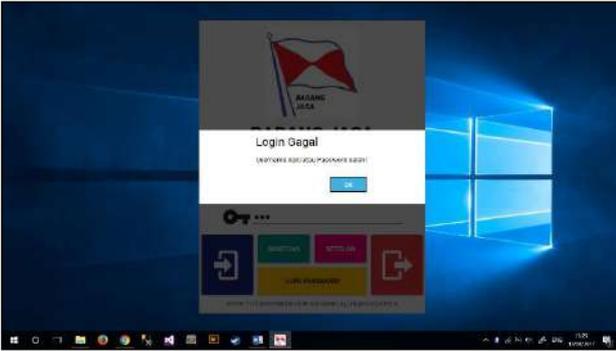


Gambar 13. Tampilan aplikasi pada resolusi 1024 x 768

Setelah dilakukan pengujian interface, kemudian dilakukan pengujian backend aplikasi. Backend aplikasi diuji dengan dua skenario. Skenario yang pertama yaitu menguji login dengan username dan password yang terdaftar di dalam database. Skenario kedua login dilakukan dengan password dan/atau username yang salah maupun tidak sesuai dengan data yang ada pada database. Gambar 14 merupakan skenario pertama jika username dan password tepat. Sedangkan pada gambar. Selanjutnya pada gambar 15 merupakan skenario kedua jika password dan/atau username yang salah maupun tidak sesuai dengan data yang ada pada database.



Gambar 14. Username dan Password user tepat



Gambar 15. Username dan/atau Password salah atau tidak terdaftar

KESIMPULAN

Setelah menguji fleksibilitas antarmuka pada layar dengan resolusi 1366 x 768, 1024 x 768 dan 1280 x 720, dibuktikan bahwa WPF dapat diatur sesuai dengan resolusi layar yang diuji. Selanjutnya, hasil tes menunjukkan bahwa backend dari aplikasi Barang Jasa berfungsi dengan baik, tes ini dilakukan dengan menggunakan query sql dengan memasukkan parameter yang diperlukan, misalnya username dan password untuk login dan mencocokkan kembali permintaan hasil dengan apa yang ditampilkan oleh aplikasi.

DAFTAR PUSTAKA

- [1] (2016) Microsoft Developer Network *Visual Studio IDE*. [Online]. Tersedia: <https://msdn.microsoft.com/en-us/library/dn762121.aspx>
- [2] Rouse, Margaret. (2012) *C# Definition*. [Online]. Tersedia: <https://searchwindevelopment.techtarget.com/definition/C>
- [3] Rouse, Margaret. (2014) *Object-oriented programming defition*. [Online]. Tersedia: <https://searchwindevelopment.techtarget.com/definition/object-oriented-programming>
- [4] (2016) Microsoft *Welcome to the .NET Framework*. [Online]. Tersedia: <https://www.microsoft.com/net/framework>
- [5] Gibilisco, Stan and Rouse, Margaret. (2012) *What is SQL Server 2012?*. [Online]. Tersedia: <http://whatis.techtarget.com/definition/SQL-Server-2012>
- [6] (2016) *Microsoft Introduction to WPF*. [Online]. Tersedia: [https://msdn.microsoft.com/en-us/library/mt149842\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/mt149842(v=vs.110).aspx)
- [7] (2016) Microsoft *XAML Overview (WPF)*. [Online]. Tersedia: [https://msdn.microsoft.com/en-us/library/ms752059\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ms752059(v=vs.110).aspx)

