

# MESIN PENCETAKAN DENGAN PENGGERAK DARI PARAMETER PROGRAM BERBASIS WINDOWS

**Dwi Sunaryono**

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember  
Kampus ITS, Jl. Raya ITS, Sukolilo – Surabaya 60111, Tel. + 62 31 5939214, Fax. + 62 31 5913804  
Email: dwi@its-sby.edu

## ABSTRAK

*Proses pencetakan merupakan alat bantu yang harus ada pada semua aplikasi terutama pada pembuatan aplikasi sistem informasi. Dua model pembuatan pencetakan yang selama ini tersedia adalah dengan pembuatan laporan langsung lewat program atau dengan pencetakan lewat alat bantu yang terpisah dengan program. Persoalan terbesar pada proses pembuatan laporan terjadi pada fleksibilitas dari isi laporan, sehingga mudah di ubah tanpa mengubah program asli (executable) atau membuka alat bantu yang dipakai. Persoalan ini dapat diatasi dengan mengubah pendekatan ke pemisahan program utama yang mengurus masalah inti dari aplikasi sedangkan pembuatan laporan pencetakan dapat ditugaskan pada mesin penggerak yang bisa di modifikasi, sehingga mudah di pelihara khususnya terjadi permintaan perubahan format laporan atau perubahan lainnya yang berkaitan dengan pembuatan laporan.*

*Pembuatan laporan dapat diatur dengan dimiripkan dengan skrip program tertentu dengan membangun fungsi generik untuk membantu mempercepat pembuatan format laporan. Tujuan pencetakan dapat diatur tergantung dari kebutuhan pelaporan yang diinginkan misalnya pencetakan lewat port LPT, lewat driver manufacturer, lewat word dan lewat excel. Khusus untuk pencetakan lewat word dapat dipilih sebagai dokumen word baru, bentuk template atau mail merge, sedangkan excel dapat memilih pencetakan lewat dokumen excel baru atau dokumen template. Tujuan pembuatan laporan ke berbagai macam pilihan sumber dapat meningkatkan kemampuan aplikasi untuk berinteraksi dengan aplikasi lain sehingga fleksibilitas dari aplikasi dapat meningkat. Setting pengaturan printer diatur sedemikian rupa sehingga mirip dengan lingkungan printer. Pengambilan data secara mudah merujuk pada data yang saat itu aktif. Untuk keperluan lain dimungkinkan membuat penampung data sendiri, kemudian di kirim ke printer. State sebelum dan sesudah pencetakan dapat di sisipkan sintak, sesuai dengan proses yang ada, seperti misalkan update status keberhasilan pencetakan, sehingga tidak dobel melakukan pencetakan. State sebelum pencetakan dapat mengecek apakah status printer telah tercetak atau belum. Pengaturan kertas, halaman tunggal atau kertas bersambung (continuous-form) dapat dihandel dengan melakukan logika pada kertas secara fisik. Pemetaan logika dengan memakai logika pemrograman seperti perulangan, percabangan dan pembuatan variabel. Semua kelengkapan skrip program sebagai parameter dari program utama menjadikan aplikasi mudah dilakukan perubahan dengan otorisasi tertentu.*

*Manajemen penyimpanan skrip parameter program untuk tiap kasus yang berbeda pada satu tempat penyimpanan dapat memudahkan pembuatan laporan. Penyimpanan bersifat sebagai sumber daya bagi pakai (resource sharing) menunjang pembuatan skrip dikerjakan oleh banyak orang pada saat yang sama, dan secara langsung dapat juga dipakai sebagai mesin pencetakan secara bersama-sama.*

**Kata kunci :** mesin Pencetakan, parameter program, penggerak.

## 1. PENDAHULUAN

Jembatan antara perangkat keras dengan perangkat lunak agar suatu mesin dapat dioperasikan adalah Sistem Operasi [3]. Mesin perangkat keras di petakan secara logika ke dalam perangkat lunak yang dapat di setting dan di program. Pemetaan ini menyebabkan manusia sebagai pemakai mesin tidak harus berhubungan dengan ke mesin perangkat keras secara langsung.

Printer sebagai perangkat keras yang bertugas mengeluarkan hasil pengolahan ke sebuah berkas fisik berupa kertas. Sistem operasi menyediakan pustaka akses ke printer tergantung dari jenis printer. Masing-masing jenis printer memiliki penggerak (*driver*) untuk setting komunikasi printer dengan

komputer. Pembuat printer menyediakan protokol aturan yang sesuai dengan sistem operasi. Printer yang telah di daftarkan di Sistem Operasi dapat dipakai dan dapat dimanipulasi untuk tujuan pembuatan laporan.

Selain pendaftaran printer secara standar dengan *driver manufacturer*, maka Sistem Operasi juga menyediakan fasilitas akses langsung ke port I/O. Perbedaan yang mencolok adalah pada pemakaian, pada printer yang didaftarkan lewat *driver manufactuer*, memiliki kelebihan pada semua fasilitas yang disediakan oleh sistem operasi, seperti misalnya printer bagi pakai (*print sharing*), sehingga satu komputer dengan printer terpasang, dapat di akses oleh banyak komputer, sedangkan

kelemahannya terletak pada kecepatan, karena harus melalui di cek terlebih dahulu (*spooling*).

Tipe yang lain adalah dengan cara mengakses langsung ke port I/O komputer. Biasanya komputer menyediakan akses langsung ke I/O, misalnya untuk printer yang terhubung ke kanal paralel (*parallel port*) LPT. Kelebihan akses langsung ke port I/O terletak pada kecepatan pencetakan, biasanya dipakai pada pencetakan laporan dengan format presisi yang tepat, misalnya pencetakan kuitansi, form pengisian dan sebagainya.

Kelemahan mencetak secara langsung ke I/O adalah pada aspek bagi pakai pencetakan (*printer sharing*). Pencetakan hanya bisa dilakukan pada printer yang melekat pada I/O komputer yang bersangkutan. Diperlukan perangkat lunak tersendiri sebagai emulasi dari kanal bayangan (*virtual port*) komputer lain, kelemahan lain adalah minimnya fasilitas pencetakan grafik, karena biasanya diperlukan konversi pengolahan tersendiri dari text ke grafik.

## 2. KELAS PRINTER

Bahasa program standar berbasis obyek (*Object Oriented Programming*) menyediakan kelas printer untuk komunikasi akses sistem operasi dengan peralatan printer. Kelas printer dapat dipakai untuk mengidentifikasi printer-printer yang tersedia sebagai koleksi printer. Kelas printer merupakan anggota kelas yang disediakan oleh microsoft / MFC (*Micorosoft Foundation Classes*)[2].

Fungsi atau metode dan atribut dari kelas printer dapat dipakai untuk komunikasi dengan peralatan printer, baik yang terhubung langsung ataupun terhubung lewat jaringan. MFC telah menyediakan standar. Bahasa tingkat tinggi seperti Visual Basic menerjemahkan MFC ke dalam kelas printer. Contoh:

```
Dim X As Printer
For Each X In Printers
    If X.Orientation = vbPRORPortrait
Then
    ' ambil printer sebagai default
    Set Printer = X
    ' hentikan pencarian
    Exit For
End If
Next
```

**Tabel 1. Daftar Metode kelas printer [4]**

No	Metode	Keterangan
1	Circle	Membuat Lingkaran
2	EndDoc	Mulai mencetak ke printer
3	KillDoc	Batalkan pencetakan
4	Line	Membuat garis/kotak
5	NewPage	Halaman Baru

6	PaintPicture	Mencetak gambar
7	Pset	Membuat titik
8	Scale	Mengatur skala
9	ScaleX,ScaleY	Mengatur skala X dan Y
10	TextHeight	Mengatur Tinggi Text
11	TextWidth	Mengatur Lebar Text

Metode kelas yang terangkum di tabel 1 harus dapat di emulasi oleh parametrik program dari luar , sehingga semua fungsi kelas printer menjadi tersedia oleh programmer.

Selain metode kelas printer juga menyediakan atribut yang berisi informasi atau state dari printer yang saat itu diaktifkan.

**Tabel 2. Daftar Atribut kelas printer [4]**

No	Atribut	Keterangan
1	ColorMode	Tentukan multi atau mono warna pencetakan
2	Copies	Tentukan jumlah pencetakan
3	Count	Koleksi jumlah printer
4	CurrentX, CurrentY	Posisi X,Y dari form/ kertas
5	Device Name	Nama peralatan printer
6	DrawMode	Tentukan mode pencetakan grafis sebagai kontrol shape atau garis
7	DrawStyle	Tipe pencetakan dari garis atau grafik
8	DrawWidth	Atur tebal garis
9	DriverName	Nama penggerak printer
10	Duplex	Keterangan printer menunjang dua sisi pencetakan
11	FillColor	Tentukan warna pengisian area grafik
12	FillStyle	Jenis arsiran area grafik
13	Font	Jenis huruf
14	FontBold, FontItalic, FontStrikeThru, FontUnderLine	Jenis atribut huruf
15	FontCount	Jumlah huruf yang tersedia pada setting printer
16	Fonts	Kumpulan nama-nama huruf yang tersedia di printer
17	FontSize	Ukuran Huruf
18	FontTransparent	Setting huruf tranparan
19	hDC	Pemegang konteks suatu obyek
20	Height,Width	Nilai tinggi dan lebar kertas

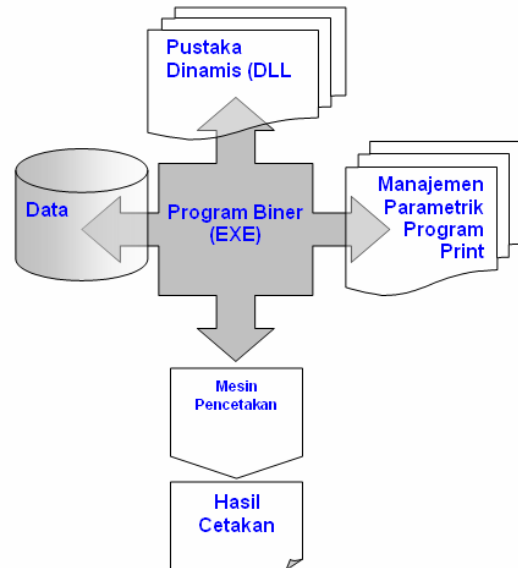
21	Orientation	Arah pencetakan
22	Page	Nomor halaman
23	PaperBin	Nilai pilihan kertas yang tersedia di printer
24	PaperSize	Ukuran kertas
25	Port	Pilihan port tempat dokumen di kirim
26	PrintQuality	Pilihan kualitas cetakan
27	RightToLeft	Nilai boolean posisi text
28	ScaleHeight,Scale Width	Nilai skala tinggi dan lebar
29	ScaleLeft, ScaleTop	Nilai skala kiri dan atas
30	TrackDefault	Nilai boolean untuk perubahan printer yang ada control panel
31	TwipsPerPixelX, TwipsPerPixelY	Nilai konversi twips dengan pixel
32	Zoom	Nilai pembesaran cetakan

Nilai atribut pada printer yang saat itu aktif terangkum dalam kelas printer yang telah di inisial. Atribut ini juga semuanya dapat di setting lewat parametrik program, sehingga secara logika kertas dapat di gambarkan sebagai suatu papan dan kita bisa menggerakkan pena, mengatur panjang papan dan sifat papan lainnya. Setelah selesai barulah papan tersebut akan menjadi dokumen yang siap dikirim ke printer. Hasil cetakan di kertas harus terlihat sama dengan setting pada papan sebelumnya.

### 3. MANAJEMEN PARAMETRIK PROGRAM PENCETAKAN

Arsitektur program yang terdiri dari program biner (*executable*) dan pustaka dinamis yang dipakai (*dynamic link library*) harus diubah menjadi parametrik program. Perubahan ini diharapkan dapat mengatasi perubahan perubahan format pencetakan secara fleksibel tanpa harus membuka program aslinya (*source code*) yang kemudian harus di kompilasi ulang [1]

Perubahan arsitekur ini dapat digambarkan pada gambar 1.



Gambar 1. Arsitekur Manajemen Parametrik Program Print dalam Program biner

Manajemen parametrik akan menyimpan aturan-aturan bagaimana suatu data dapat di cetak. Penanganan semua sumber daya yang ada disimpan dalam aturan tertentu. Aturan tersebut juga meliputi bagaimana hubungan dengan database, peralatan lain selain juga I/O misalnya printer yang dipakai untuk pembuatan laporan.

Sintak program pencetakan di miripkan pada bahasa program yang di terjemahkan ke dalam skrip. Mesin akan menerjemahkan skrip tersebut dan melakukan eksekusi sesuai dengan urutan skrip yang tersimpan.

Penyimpanan skrip di atur sedemikian rupa sehingga tiap proses di handel oleh kode tertentu. Dengan demikian semua proses dapat di simpan dalam file yang sama. File penyimpanan harus dapat di pakai bersama sehingga menunjang aspek bagi pakai, selain itu juga dapat dipakai untuk membuat setting laporan secara bersama-sama.

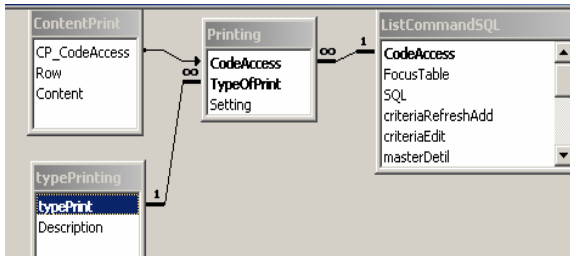
Operator komputer tidak perlu tahu adanya proses internal pencetakan, kecuali menekan tombol yang telah disediakan. Mesin akan menginterpretasikan semua skrip yang berada di kode tersebut sampai ditemukan perintah untuk mengirim dokumen ke printer.

Ketika hasil diketahui kemudian terjadi kesalahan tertentu pada hasil cetakan, maka programmer secara bersamaan (*on line*) tanpa harus menghentikan pekerjaan operator, dapat memperbaikinya. Perbaikan ini akan langsung berpengaruh ketika operator akan mencetak lagi.

### 4. PERANGKAT LUNAK MESIN PENCETAKAN

Mesin pencetakan merupakan sub modul dari suatu aplikasi lengkap yang generik dan dapat dipakai untuk keperluan pengolahan data. Jenis

pencetakan dapat di pilih sesuai tujuan. Penyimpanan aturan ini secara manajemen dapat terlihat pada tabel- tabel yang diperlihatkan pada gambar 2.



**Gambar 2. Relasi tabel Manajemen Penyimpanan Program Parametrik Program**

Tipe print dapat dipilih berdasarkan keinginan hasil cetakan. Daftar pilihan dapat di lihat pada tabel 3.

**Tabel 3 Jenis tipe tujuan Pencetakan**

typePrint	Description
0	LPT
1	Direct With driver from manufacture
2	To Word
3	To Word with mail merge
4	To Excell
5	To Excell with template
6	To Word with template

Setiap tipe tujuan pencetakan memiliki aliran eksekusi yang berbeda. Mesin membedakan dari kode angka typePrint untuk menentukan algoritma yang akan di jalankan.

**4.1. ALGORITMA PRINT MELALUI PORT**

Pencetakan langsung melalui port dapat diartikan sebagai hubungan kelas form dengan sistem operasi. Sebelum dipakai, maka port harus dibuka dan baru kemudian data di tranfer secara langsung. Pengaturan posisi pencetakan harus melalui karakter ASCII standar, misal memakai tombol spasi atau tab. Pindah baris dapat memakai karakter Line Feed atau Carriage Return.

Pembuatan garis atau kotak dapat juga dilakukan dengan menggabungkan kontrol perulangan dan karakter ASCII tambahan (*extended ASCII*).

Algoritma ini adalah :

- 0 : proses print dari status proses sebelumnya
- 1: ambil kode print
- 2: jika tidak ada, maka print tidak dilakukan
- 3: buka port I/O untuk pencetakan
- 4: mulai dari baris terkecil sampai selesai kerjakan no 5
- 5: ambil isi skrip
- 6: terjemahkan skrip
- 7: cek apakah terakhir jika ya kerjakan no 8
- 8: tutup port
- 9: kembali ke status sebelumnya

Pencetakan melalui port ini cukup cepat tanpa harus ada penundaan dalam wadah antrian (*spooling*).

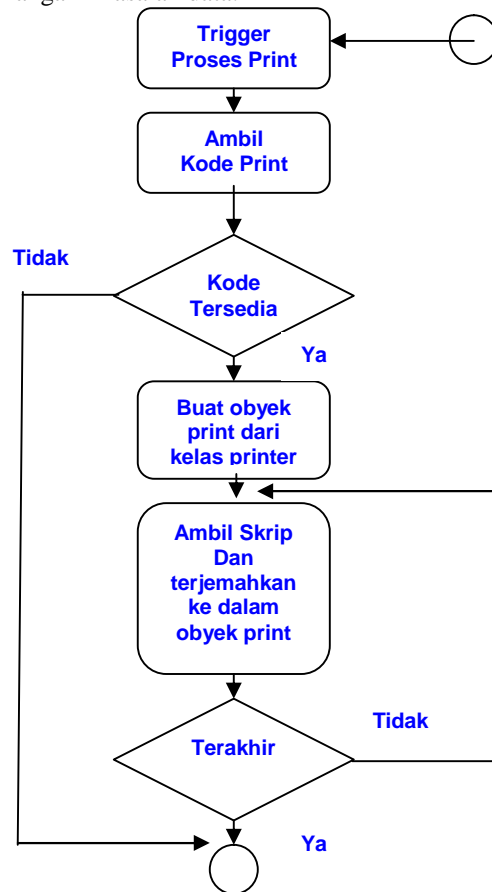
**4.2. ALGORITMA PRINT MELALUI DRIVER**

Aliran eksekusi dari algoritma dapat gambarkan pada gambar 3.

Dari suatu state program utama, diperlukan suatu trigger tertentu misalnya penekanan tombol print untuk mulai proses pencetakan dengan menggunakan mesin ini. Kemudian di cek apakah di dalam file penyimpanan parametrik program prin kode tersebut ada, jika tidak maka mesin tidak melakukan kegiatan atau kembali ke status sebelumnya dengan menampilkan pesan bahwa kode tidak ditemukan.

Ketika kode ditemukan maka sesungguhnya disitu terdapat sekumpulan data yang berisi skrip pada kelas printer. Obyek printer harus dibuat terlebih dahulu sebagai bentuk instan dari kelas printer. Setelah terbentuk obyek printer, maka mesin akan melakukan penerjemahan setiap skrip yang masuk.

Mesin tidak hanya menangani masalah obyek printer, namun juga dapat melakukan proses penanganan pada kelas lain. Biasanya dalam sebuah pengolahan informasi terdapat kelas form, yang dapat di manipulasi dengan menggabungkan dengan pada kelas printer, selain itu khusus untuk penanganan data, maka tersedia kelas database yang khusus menangani masalah data.



**Gambar 3. Arus eksekusi dari Perangkat lunak mesin pencetakan**

Komunikasi beberapa kelas dapat diatur dalam mesin ini, sehingga lebih praktis dan fleksibel untuk membuat laporan pencetakan. Obyek dari instan kelas dapat di tutup dan dibuka saat waktu jalan (*run time*) program.

Kontrol program dapat diterjemahkan oleh mesin sebagaimana halnya struktur kontrol bahasa program umumnya. Perulangan dapat dikontrol dengan pembuatan variabel secara instan.

Contoh skrip pencetakan :

```
<<for x=0 to 10 : r.print "data
print":next>>
```

Kelas form diwakili oleh huruf r yang diartikan sebagai form dan kelas print sebagai anggota dari kelas form. Hasil dari skrip sederhana diatas adalah mencetak tulisan "data print" sebanyak 10 kali.

#### 4.3. ALGORITMA PRINT MELALUI MS OFFICE

Perangkat lunak mesin pencetakan di sediakan jembatan hasil ekspor ke keluarga msoffice sbagai perangkat lunak yang populer. Kelas keluarga msoffice seperti excel dan word di ikutkan ke dalam mesin, sehingga dapat di manipulasi dalam bentuk skrip program.

Algoritma pencetakan ini adalah :

- 0 : proses print dari status proses sebelumnya
- 1: ambil kode print
- 2: jika tidak ada, maka print tidak dilakukan
- 3: buat kelas msoffice
- 4: mulai dari baris terkecil sampai selesai kerjakan no 5
- 5: ambil isi skrip
- 6: terjemahkan skrip
- 7: cek apakah terakhir jika ya kerjakan no 8
- 8: tutup kelas msoffice
- 9: kembali ke status sebelumnya

Kelas msoffice dapat di ambil sebagai komponen pustakan dinamis (DLL). Fasilitas ini hanya tersedia jika sistem operasi telah terinstal keluarga msoffice. Pilihan dapat dipakai untuk membuat file baru atau memanfaatkan file yang telah ada, bisa juga memakai fasilitas pencetakan masal (*mail merge*).

#### 5. UJI COBA

Uji coba dilakukan pada semua bentuk pencetakan.

##### Ujicoba 1 :

Print dengan fasilitas port

Skrip :

```
<<r.OpenLPT "1": r.txt(0):for i=0 to 5:r.PrintLPT "data uji
coba mencetak ke : " & i :next:r.CloseLPT:>>
```

CodeAccess	TypeOfPrint	Setting
USS	6	<<Template=templateWordyyy.Doc>>
TDST	3	<<Template=templateCustomerJAWABANDTL.doc>>

Keterangan :

Form berisi txt(0) adalah "DATA PELANGGAN"

Hasil :

**DATA PELANGGAN**  
**Data ujicoba mencetak ke 0**  
**Data ujicoba mencetak ke 1**  
**Data ujicoba mencetak ke 2**  
**Data ujicoba mencetak ke 3**  
**Data ujicoba mencetak ke 4**

##### Ujicoba 2 :

Print dengan fasilitas driver manufacturer

Skrip :

```
<<r.xprinter.Font.size=12:r.xprinter.Font.Bold=true:
r.printxy 8200,500, r.txt(2).text:
r.printxy 3150,550, "SURABAYA SELATAN":
r.printxy 1700,750, "UPPTR NGAGEL":r.xprinter.EndDoc>>
```

Hasil :

**SURABAYA SELATAN**  
**UPPTR NGAGEL**  
**BEECA20040406**

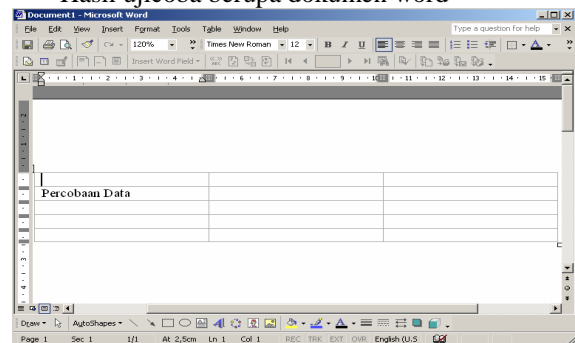
##### Ujicoba 3 :

Print dengan fasilitas ekspor ke ms word

Skrip :

```
<<r.oDoc.Select:set r.odocSelect = r.oWord.Selection:
r.oDoc.Tables.Add r.odocSelect.Range, 5, 3>>
<<r.oDoc.Tables(1).Cell(2, 1).Range = "Percobaan Data">>
```

Hasil ujicoba berupa dokumen word



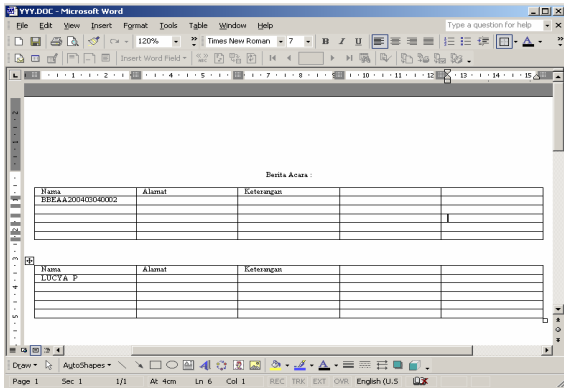
Gambar 4. Hasil pencetakan dengan msWord

##### Ujicoba 4 :

Print dengan fasilitas ekspor ke ms word dengan file template yyy.doc

CP	CodeAccess	Row	Content
USS		2	<<r.odoc.Tables(2).Cell(2,1).range=r.txt(2)>>
USS		1	<<r.odoc.Tables(1).Cell(2,1).range=r.txt(1)>>

Hasil ujicoba dokumen word dengan template



Gambar 5. Hasil pencetakan msWord template

**Ujicoba 5 :**

Print dengan fasilitas expor ke ms word dengan mail merge

Skip :

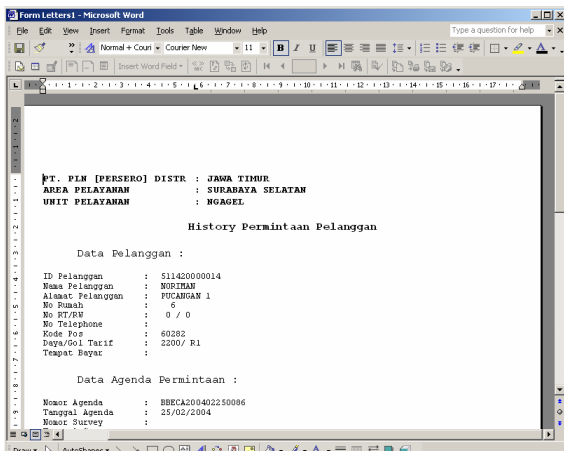
CodeAccess : HP

TypeOfPrint : 3

Setting :

<< Template=\templateCustomer\History Permintaan Pelanggan.doc>>

Hasil ujicoba dokumen word dengan mailmerge



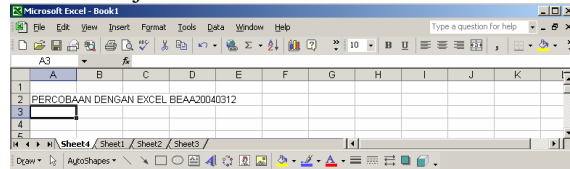
Gambar 6. Hasil pencetakan dengan mail merger msWord

**Ujicoba 6 :**

Print dengan fasilitas expor ke excel

<<r.oSheet.Cells(1,1).Value = r.txt(1):r.oSheet.Cells(2,1).value="PERCOBAAN DENGAN EXCEL" & r.txt(2)>>

**Hasil ujicoba dokumen excel**



Gambar 7. Hasil pencetakan dengan excel

**Ujicoba 7 :**

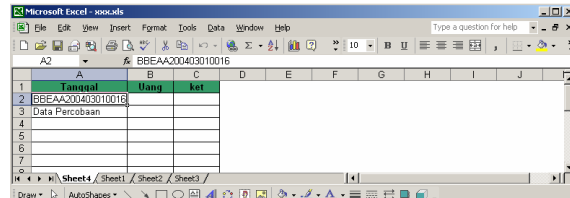
Print dengan fasilitas expor ke file template excel

CodeAccess: PK

TypeOfPrint: 5

Setting : <<Template=\templateExcel\xxx.xls>>

Hasil ujicoba dokumen template file excel



Gambar 8. Hasil pencetakan dengan file template excel

**6. KESIMPULAN**

Dari hasil uji coba yang telah dilakukan dan perbandingan beberapa kasus didapatkan beberapa kesimpulan sebagai berikut :

1. Mesin pencetakan dapat berfungsi dengan baik sesuai tujuan yakni dapat mencetak laporan pada berbagai macam tujuan
2. Mesin dapat menerjemahkan instruksi yang diberikan lewat parametrik program yang tersimpan dalam file MsAccess
3. Penyimpanan parametrik program pencetakan sangat membantu fleksibilitas pengubahan pola format laporan

**7. DAFTAR PUSTAKA**

1. Microsoft Foundation Class, Microsoft Help OnLine, 2001
2. MSDN Library Visual Studio 6, MSDN , 2001
3. Stalling,William, "Operating System", Prentice Hall, 2003
4. Sunaryono, Dwi. "Pelacakan kembali kontrol-kontrol obyek pada penambahan obyek secara dinamis Berbasis visual basic untuk studi kasus analisa hirarki tugas", JUTI, Volume 1, Nomor 1, Juli 2002 : 12 – 19