



MEMBANGUN APLIKASI AUTOGENERATE SCRIPT KE FLOWCHART UNTUK Mendukung BUSINESS PROCESS REENGINEERING

¹Muslim Setyo Rejeki, ²Ali Tarmuji(0014107301)

^{1,2} Program Studi Teknik Informatika
Universitas Ahmad Dahlan

Prof. Dr. Soepomo, S.H., Janturan, Umbulharjo, Yogyakarta 55164

²Email: alitarmuji@tif.uad.ac.id

ABSTRAK

Pesatnya persaingan antarperusahaan, menjadikan salah satu motivasi perusahaan untuk melakukan perbaikan. Bisnis process reengineering merupakan usaha untuk memperbaiki dan meningkatkan kinerja dari semua bagian atau divisi yang ada dalam proses pelayanan kepada pelanggan, efisiensi proses transaksi. Proses reengineering merupakan bagian dari bisnis process reenginerring yang mengkaji ulang dari sistem yang sudah ada. Dalam Bisnis process reenginerring yang diterapkan adalah untuk membantu analis maupun programmer dalam memahami aplikasi yang sudah ada. Dimana terdapat kendala disaat melakukan reenginerring pada sistem yang sudah ada, dimana membutuhkan pemahan lebih untuk dapat melakukan pembenahan pada sistem yang sudah ada. Pembacaan script atau kode sumber pada aplikasi yang sudah ada menjadi salah satu kendala yang seringkali terjadi khususnya oleh analis.

Penelitian dimulai dengan mengumpulkan referensi yang berkaitan dengan pembuatan aplikasi konversi. Metode pengumpulan data yaitu dengan metode wawancara pada analis, observasi pada suatu instansi, dan studi pustaka. Metode penelitian yang digunakan dengan analisa kebutuhan dan perancangan sistem yang meliputi perancangan proses dalam pembuatan aplikasi, perancangan grammar yang digunakan sebagai aturan baku dalam pembuatan aplikasi, perancangan scanner yang digunakan dalam mengetahui token-token dari script yang dimasukkan untuk membantu proses parsing, perancangan parser yang mengubah dari token-token yang didapat menjadi aturan untuk pembuatan flowchart, perancangan simbol flowchart dari hasil parser, dan perancangan interface. Pada langkah terakhir dengan melakukan pengujian dari aplikasi yang telah dibuat.

Penelitian yang dibuat menghasilkan sebuah aplikasi yang berfungsi melakukan autogenerate script menjadi flowchart pada browser yang membantu analis memahami script dari aplikasi dalam mendukung bisnis process reenginerring.

Kata Kunci: script, business process reenginerring, flowchart, autogenerate.

1. PENDAHULUAN

Business process reengineering merupakan pemikiran ulang secara fundamental dan desain ulang secara radikal terhadap sistem bisnis untuk mendapatkan

peningkatan dramatis dalam pengukuran kinerja kritikal dan kontemporer seperti biaya, kualitas, pelayanan, dan kecepatan [8]. Konsep ini memberikan gambaran bahwa untuk mencapai peningkatan yang dramatis (mencapai lompatan dalam kinerja dan tidak kemajuan sebagian-sebagian atau berkelanjutan) untuk perubahan menjadi sebuah sistem yang baru dan mendasar, yang berbeda dengan sistem yang lama. Dengan demikian, perlu mendapatkan akar atau sumber dari segala sesuatu dan menemukan secara komplit cara atau jalan baru untuk melakukan perbaikan, baik dari manajemen puncak maupun dari pemilik perusahaan, dalam mengkaji kembali dalam berbagai aspek fundamental dari bisnis yang dijalankan. Dengan adanya ini, diketahui bahwa proses bisnis memiliki peranan penting dalam meningkatkan kinerja dari perusahaan. Kemudian setelah melakukan pembenahan dalam hal proses bisnis yang ada, mengakibatkan banyaknya permintaan dari pelanggan melebihi dari sebelum-sebelumnya dan pasar persaingan akan lebih mudah untuk dapat dimenangkan. Baik dari sisi terciptanya produk yang siap bersaing dipasaran dengan harga dan kualitas yang memuaskan.

Terdapat hal mendasar dalam BPR (*Business process reengineering*), dimana pemilik perusahaan atau manajemen harus memiliki cara tepat dan efisien untuk meningkatkan kinerja perusahaan dari segala aspek yang menjadi penentu untuk kemajuan dari perusahaan. Dalam *Business process reengineering* yang diterapkan hanya mencakup pada teknologi yang ditangani oleh analis maupun programmer. Saat perkembangan teknologi yang sangat pesat saat ini, dapat berperan dalam meningkatkan kinerja perusahaan dalam bidang industri dan bisnis. Dengan demikian, pemanfaatan IT sangatlah diperlukan untuk mempermudah perusahaan khususnya dalam persaingan dengan perusahaan lainnya.

Memaksa banyak perusahaan untuk memigrasikan *legacy sistem* ke dalam sistem baru. Proses migrasi tersebut adalah *reengineering*. Bentuk sistem baru tersebut mengaplikasikan pada perubahan paradigma yang mampu menangani perancangan arsitektur. Saat manajer memodifikasi aturan-aturan bisnis untuk mencapai keefektifan dan komposisi yang lebih besar, perangkat lunak harus tetap berjalan maju. Artinya penciptaan sistem berbasis komputer yang besar berarti memodifikasi atau membangun aplikasi yang sudah ada sehingga menjadi kompeten untuk memenuhi kebutuhan bisnis pada masa yang akan datang.

Banyak perusahaan yang telah menggunakan jasa perkembangan teknologi saat ini untuk membantu meningkatkan kinerja perusahaan, akan tetapi sistem yang sudah ada tidak dapat memenuhi kebutuhan sesuai dengan proses bisnis yang ada, terlebih permintaan dari pelanggan yang selalu menyesuaikan dengan tren saat itu, menjadikan perusahaan harus mengkaji ulang sistem yang sudah ada untuk memenuhi kebutuhan-kebutuhan yang semestinya. Dengan demikian, sistem yang ada dilakukan rekayasa ulang untuk disesuaikan dengan kebutuhan-kebutuhan yang mengalami perubahan dari sebelumnya. Proses rekayasa ulang proses bisnis merupakan usaha untuk memperbaiki dan meningkatkan kinerja dari semua bagian atau divisi yang ada dalam proses pelayanan kepada pelanggan, efisiensi proses transaksi. Tahap rekayasa ulang meliputi : pendefinisian bisnis, identifikasi proses, evolusi proses, desain proses, *prototyping*, penyaringan beserta instalasi, implementasi *reengineering*, dan evaluasi *reengineering* [8]. Dari tahapan tersebut, terdapat proses *reengineering* yang mengkaji ulang dari sistem yang sudah ada, kemudian dirubah ke dalam bentuk alur dari sistem yang dapat dipelajari kemudian dianalisis kembali. Terdapat kendala disaat melakukan rekayasa ulang pada sistem

yang sudah ada tersebut, dimana dibutuhkan pemahaman lebih untuk dapat melakukan pembenahan pada sistem yang sudah ada. Pembacaan *script* atau kode sumber pada aplikasi yang sudah ada menjadi salah satu kendala yang seringnya terjadi khususnya oleh analis. Dalam memahami aplikasi, meski terkadang sudah terdapat dokumentasi dari sistem yang ada, akan tetapi kurang lengkap maupun tidak adanya dokumentasi dalam beberapa kasus menjadikan sulitnya bagi analis dalam memahami aplikasi yang ada. Perlu adanya suatu cara yang memudahkan analis dalam memahami isi dari *script* yang ada.

2. KAJIAN PUSTAKA

Pada penelitian ini digunakan beberapa referensi yang berhubungan dengan materi penelitian. Referensi diambil dari contoh-contoh hasil penelitian sebelumnya. Dalam penelitian oleh Sunandar Hariyanto, 2008 yang berjudul "**Pengembangan Perangkat Lunak Untuk Mendukung *Business Process Reengineering* (BPR) Di Penerbit Adicita Grup**" membahas mengenai Business Process Reengineering yang diterapkan pada Penerbit Adicita Grup. Dalam penelitian tersebut, menjelaskan mengenai penerbit Adicita Grup seiring bertambahnya waktu selalu terjadi perubahan dalam proses bisnis yang berjalan. BPR membantu pada penerbit dalam melakukan pembenahan pada bisnis proses yang berkembang [8].

Selain itu, Penelitian yang akan diambil ini juga mengacu pada penelitian yang dilakukan oleh Abdul Basit, 2006 yang berjudul "**Pembuatan Kompiler Dengan Metode *Recursive Descent Parser***" yang membahas mengenai pembuatan kompiler yang menggunakan metode Recursive Descent Parser. Dimana dalam penelitian yang dibuat membuat kompiler yang dalam pembuatannya mencakup scanner dan parser. Dengan hasil akhir pada penelitian adalah berupa file exe [4].

Dengan melihat dari kedua penelitian terdahulu diatas maka akan dibuat penelitian yang berjudul "**Membangun Aplikasi *Autogenerate Script Ke Flowchart* Untuk Mendukung *Business Process Reengineering***". Dalam penelitian ini akan membahas mengenai pembuatan aplikasi konversi *script* menjadi *flowchart* yang digunakan untuk mendukung pada *Business Process Reengineering*.

2.1. Teori *Business Process Reengineering*

Berikut definisi mengenai *business process reengineering* menurut beberapa pakar. *Business process reengineering* merupakan pemikiran ulang secara fundamental dan desain ulang secara radikal terhadap sistem bisnis untuk mendapatkan peningkatan dramatis dalam pengukuran kinerja kritical dan kontemporer seperti biaya, kualitas, pelayanan, dan kecepatan [7]. *Reengineering* adalah perubahan yang cepat dan radikal dalam merancang ulang strategi, nilai tambah proses bisnis dan sistem, aturan dan struktur organisasi yang mendukung kegiatan bisnis, dan untuk mengoptimalkan *workflow* dan produktivitas di dalam suatu organisasi [7].

2.2. Teori Bahasa Dan Automata

Otomata berasal dari kata *automatic* yang artinya bekerja dengan sendirinya. Secara istilah, otomata adalah ilmu yang mempelajari mesin abstrak yang dapat menerima masukan secara sekuensial dan dapat menghasilkan keluaran. Setiap

otomata memiliki mekanisme untuk membaca masukan dari awal hingga akhir. Otomata juga dapat dirancang untuk menghasilkan keluaran, menyimpan sementara simbol dari alfabet dan memanipulasi isi dari sel simpanan. Otomata mempunyai unit kendali yang memiliki berhingga status dan mampu mengubah status tersebut [9]. Secara umum, otomata dapat digolongkan menjadi dua, yaitu *accepter* dan *transducer*. *Accepter (recognizer)* adalah otomata yang akan membuat keputusan tentang diterima atau tidaknya suatu masukan.

2.3. Flowchart

Flowchart merupakan penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. Bagan alir (flowchart) adalah bagan (chart) yang menunjukkan alir (flow) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi.

3. METODE PENELITIAN

Subjek penelitian ini adalah pembuatan aplikasi untuk mengkonversi *script* menjadi *flowchart* untuk mendukung *business process reengineering*.

3.1. Metode Pengumpulan Data

Metode pengumpulan data merupakan suatu metode atau cara untuk mendapatkan data-data yang dibutuhkan. Metode yang digunakan adalah

3.1.1. Observasi

Metode pengumpulan data yang diperoleh berdasarkan pengamatan langsung pada objek penelitian untuk memperoleh data yang diperlukan dan gambaran mengenai hal-hal yang dibutuhkan. Dalam hal ini yang diamati adalah sulitnya analisis dalam memahami *script* dari aplikasi yang ada untuk mendukung *business process reengineering*.

3.1.2. Wawancara

Metode pengumpulan data yang dilakukan dengan mengadakan tanya jawab kepada pegawai di Stikes Aisyiyah sebagai analisis maupun programmer sehingga dapat diketahui permasalahan-permasalahan yang terjadi dalam pembuatan aplikasi dalam mendukung *business process reengineering*.

3.1.3. Studi Pustaka

Studi pustaka adalah melakukan pengumpulan data dari buku-buku atau literatur yang telah disusun. Pada umumnya, buku-buku tersebut berisi landasan teoritis yang tersusun secara sistematis. Studi pustaka lebih banyak diperlukan dalam perancangan.

3.2. Analisis Kebutuhan

Analisis data merupakan satu tahapan dalam menganalisa data yang telah didapatkan baik itu secara teknis maupun non teknis. Adapun data secara teknis adalah data manual yang memiliki informasi mengenai masalah yang menjadi objek penelitian, sedangkan data non teknis adalah data-data yang digunakan sebagai penunjang untuk pertimbangan dalam membangun aplikasi. Dalam analisis kebutuhan ini, terbagi menjadi dua yaitu analisis kebutuhan sistem dan analisis

kebutuhan user. Analisis kebutuhan sistem menjelaskan rancangan dari sistem yang akan dibuat, baik dari segi aturan maupun kemampuan dari sistem. Adapun analisis kebutuhan user menjelaskan mengenai fasilitas sistem yang berhubungan dengan user tersendiri. User yang dimaksud adalah analis maupun programmer yang akan menggunakan aplikasi ini.

3.3. Perancangan Sistem

1. Perancangan Proses
2. Perancangan *Grammar*
3. Perancangan *Scanner*
4. Perancangan Parser
5. Perancangan Simbol *Flowchart*
6. Perancangan *Interface*

3.4. Implementasi

Pada tahapan ini merupakan hasil jadi dari aplikasi yang telah dibuat, yang merupakan gambaran-gambaran dari aplikasi. Pada tahapan ini, dapat dilihat dari hasil *grammar* hingga *parser* yang diimplementasikan dalam browser yang berbentuk *flowchart*.

3.5. Pengujian

Setelah aplikasi siap digunakan, langkah berikutnya adalah melakukan uji coba atau pengujian yang bertujuan untuk mengoptimalkan kinerja perangkat lunak.

4. HASIL DAN PEMBAHASAN

4.1 Analisis Kebutuhan

Aplikasi yang akan dibuat terdapat beberapa hal yang harus diperhatikan, baik dari sisi sistem maupun dari sisi user. Adapun hal-hal yang perlu diperhatikan, yaitu:

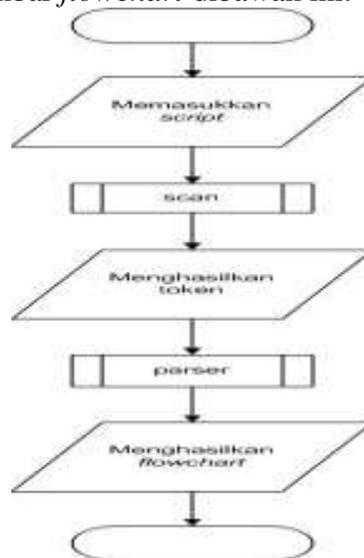
1. Kebutuhan Sistem
 - a. Dalam pembuatan aplikasi ini mencakup beberapa elemen yang digunakan seperti *scanner*, dan *recursive descent parser*.
 - b. Studi kasus dalam bahasa pemrograman yang digunakan yaitu PHP, sehingga aplikasi ini akan membaca kode sumber php yang dimasukkan kemudian melakukan konversi menjadi *flowchart* sesuai dengan kode sumber yang dimasukkan.
 - c. Dalam aplikasi yang akan dibuat, hanya dapat melakukan konversi dari kode sumber yang bebas *error* menjadi *flowchart*, dengan demikian apabila terdapat kode sumber yang *error* maka akan muncul pesan peringatan di browser.
 - d. Dalam pembuatan aplikasi konversi ini, tidak semua kode sumber dapat dikonversi namun hanya kode sumber yang meliputi *if-else*, *do-while*, *for*, dan prosedur (*function*). Aplikasi ini tidak dapat menangani *class* dan yang lainnya.
 - e. Pada listing program yang dimasukkan, menghasilkan *program flowchart*, dimana hasil *flowchart* tersebut merupakan alur-alur yang ada pada kode sumber yang dimasukkan.
2. User

Adapun pengguna dari aplikasi ini meliputi seorang analis, programmer maupun manajer IT yang ingin mengetahui alur dari kode sumber yang ditampilkan dalam bentuk *flowchart* dalam mendukung *business process reengineering*.

4.2 Perancangan Sistem

1. Perancangan Proses

Perancangan proses dari sistem perangkat lunak ini digunakan untuk mengolah *script* yang dimasukkan. Dalam gambaran dari aplikasi yang dibuat, dapat diketahui alur dari aplikasi dengan gambar *flowchart* dibawah ini.



Gambar 1. *Flowchart* Aplikasi

Dari gambar *flowchart* di atas, diketahui bahwa dalam aplikasi yang dibuat membutuhkan masukan berupa *script* (php). *Script* tersebut akan dibaca karakter demi karakter yang akan menghasilkan token-token sesuai dengan tipenya masing-masing. *Flowchart* diatas menggambarkan alur pembuatan aplikasi secara menyeluruh, dimana terdapat beberapa komponen di dalamnya baik *scanner* maupun parser.

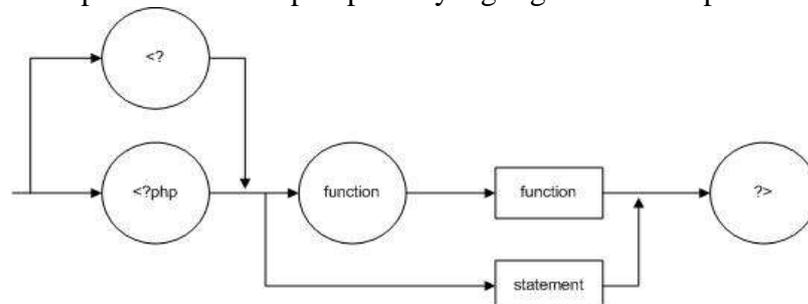
2. Perancangan Grammar

Dalam perancangan sistem membutuhkan beberapa analisa dari *grammar* yang di dalamnya merupakan suatu analisis dari suatu implementasi yang saling berhubungan, sehingga aturan produksi dalam bentuk BNF pada tabel berikut:

Tabel 1. *Grammar*

| |
|---|
| <pre><main> ::= { TBUKA TBUKA2 (TFUNCTION <function_> <statement>) TTUTUP }</pre> |
| <pre><statement> ::= { <if_> <loop_> (TECHO TRETURN PRINT TPRINT_R TPRINT_F) (<string_> TIDENTIFIER) TIDENTIFIER ((TSAMADENGAN <operator>) <penugasan_>) ((TIDENTIFIER TLP) <exp> <string_>) TLCB <statement> TRCB }</pre> |
| <pre><exp> ::= TIDENTIFIER { TIDENTIFIER }</pre> |
| <pre><string_> ::= (TPETIKTUNGGAL TPETIKGANDA) <exp> (TPETIKTUNGGAL TPETIKGANDA)</pre> |
| <pre><operator_> ::= TTAMBAH TKURANG TKALI TBAGI TSISA TAND TOR TXOR TKONKATENASI</pre> |
| <pre><penugasan_> ::= [<operator_>] TSAMADENGAN [<exp>]</pre> |

Pada tahap selanjutnya yaitu proses *parser* yang dilakukan. Token yang dihasilkan setelah melakukan *scanner*, maka akan dibaca oleh *parser* untuk dilakukan pemetaan pada token. Adapun parser yang digunakan berupa:



Gambar 3. Diagram sintaks Main

Pada diagram sintaks di atas berdasarkan pada grammar `<main> ::= { TBUKA | TBUKA2 (TFUNCTION <function_> | <statement>) TTUTUP }`. Pada TBUKA merupakan hasil dari `<?php` yang menjadi acuan pada script php. Dalam php terdapat dua cara pembukaan sintaksnya baik `<?php` maupun `<?`, keduanya dapat digunakan sesuai dengan keinginan. Setelah pembukaan pada script, dapat bertemu dengan function maupun statement. Diakhir dari grammar, berupa penutup pada script php yaitu `?>` (TTUTUP).

5. Perancangan Simbol *Flowchart*

Pada proses pembuatan *flowchart*, akan disimpan pada antrian dari setiap hasil token kemudian setelah semua selesai dalam pembuatannya barulah dilakukan pembuatan gambar semuanya. Sehingga gambar-gambar yang akan dibuat dilakukan penampungan terlebih dahulu.

6. Perancangan Interface

Dalam perancangan ini, terbagi menjadi beberapa halaman yang digunakan. Halaman yang digunakan berupa halaman utama dengan halaman panduan maupun aturan.

4.3 Implementasi

Pada bagian implementasi ini, merupakan bagian penting dari aplikasi. Dimana parsing yang dilakukan dimulai dari function main ini. Fungsi main ini akan membaca setiap masukan dari *script* hingga akhir. Dari hasil pembacaan, maka dari token-token yang ada, kemudian dilakukannya parsing.

Dalam proses parsing melakukan pemetaan dari hasil scanner yang mendapatkan token kemudian dilakukan pengantrian melalui stack agar dalam proses penggambaran pada browser berjalan. Pada proses pengantrian, yang didapatkan lebih dahulu akan diantrikan dalam stack. Kemudian setelah itu seluruh yang antrian dilakukan penggambaran pada browser.

4.4 Pengujian

Dalam melakukan pengujian pada aplikasi *autogenerate*, digunakan 2 metode yang dilakukan yaitu *black box* dan *alpha test*. Berdasarkan *prosentasi* penilaian tersebut, dapat disimpulkan bahwa pada aplikasi *autogenerate* tersebut telah sesuai dengan pengguna (*user friendly*).

5. DAFTAR PUSTAKA



- [1]. Abdul Kadir. 2002. *Dasar Pemrograman Web Dinamis Menggunakan PHP*. Yogyakarta: Andi.
- [2]. Azwar, 2008, "Aplikasi pengkonversi file teks (.rtf dan .txt) ke file suara (.wav)", Skripsi S-1, Universitas Ahmad Dahlan, Yogyakarta.
- [3]. Bambang Hariyanto, Ir., MT. 2004, *Teori Bahasa, Otomata Dan Komputasi Serta Terapannya*. Bandung: Informatika Bandung.
- [4]. Basit, Abdul, 2006, "Pembuatan Kompiler Dengan Metode Recursive Descent Parser". Skripsi S-1, Universitas Ahmad Dahlan, Yogyakarta.
- [5]. Darmawan, Cristina, 2008, "Generator Flowchart & Menghitung LOC Dalam Program Visual Foxpro", Skripsi S-1, Universitas Kristen Duta Wacana, Yogyakarta.
- [6]. Davenport, Thomas H, *Prosess Inovation, Reengineering work through informastion technology*, Harvard Business school press, 1993
- [7]. Hammer, Michel, James Champy, *Reengineering the corporation*, Harper business, 1993.
- [8]. Hariyanto, Sunandar, 2007, "Pengembangan Perangkat Lunak Untuk Mendukung Business Process Reengineering(BPR) Di Penerbit Adicita Grup". Skripsi S-1, Universitas Ahmad Dahlan, Yogyakarta.
- [9]. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, 2007. *Teori Bahasa Dan Otomata*. Yogyakarta : Andi.
- [10]. Sugiarto, Bambang, 2012, "Aplikasi Penterjemah Bahasa Indonesia Ke Bahasa Lampung", Skripsi S-1, Universitas Ahmad Dahlan, Yogyakarta.
- [11]. Utdirartatmo FIRRAR. 2005. *Teknik Kompilasi*. Yogyakarta: Graha Ilmu