

PENGEMBANGAN APLIKASI PENYUSUN JADWAL PELAJARAN SECARA OTOMATIS

Erwin Herdiansyah¹, Rinda Cahyana MT²

Jurnal Algoritma
Sekolah Tinggi Teknologi Garut
Jl. Mayor Syamsu No 1 Jayaraga Garut 44151 Indonesia
Email : jurnal@sttgarut.ac.id

¹1106038@sttgarut.ac.id
²rindacahyana@sttgarut.ac.id

Abstrak - Jadwal pelajaran merupakan hal yang penting dalam dunia pendidikan. Setiap proses pembelajaran mengikuti jadwal pelajaran yang telah disediakan agar tidak terjadi bentrok saat melakukan pengajaran. Pada pelaksanaannya, masih terdapat jadwal pelajaran yang bentrok dengan jadwal pelajaran yang lain. Metode pengembangan perangkat lunak yang digunakan adalah Unified Software Development Process dan untuk pemodelan menggunakan Unified Modeling Language. Sedangkan untuk pembuatan antarmuka menggunakan Windows Presentation Foundation dan untuk basis datanya menggunakan phpMyAdmin. Pengujian aplikasi dilakukan dengan menggunakan metode black box yang terdiri dari Usability test dan user satisfaction. Hasil dari penelitian ini berupa aplikasi desktop yang berfungsi untuk menyusun jadwal pelajaran dan mencegah bentrok antar jadwal pelajaran secara otomatis untuk melengkapi fitur pada penelitian sebelumnya.

Kata Kunci – Aplikasi desktop, Jadwal Pelajaran, Unified Software Development Process.

I. PENDAHULUAN

Penjadwalan pelajaran merupakan pekerjaan yang rutin dalam sistem akademik di sebuah sekolah yang dilakukan setiap menghadapi semester baru ataupun tahun ajaran baru, termasuk di Sekolah Menengah Atas/Sederajat. Pada pelaksanaannya, Masih terdapat sekolah yang menyusun jadwal tersebut dengan menggunakan alat tulis biasa. Hal ini membuat sering terjadinya bentrok dalam jadwal pelajaran karena dalam penyusunannya hanya mengandalkan naluri dari staf penyusun jadwal saja. Ada juga yang telah menggunakan aplikasi Microsoft Excel dalam pembuatan jadwal pelajaran tersebut, akan tetapi aplikasi tersebut masih belum optimal untuk menyelesaikan masalah pembuatan jadwal ini karena pada pelaksanaannya staf pembuat jadwal masih tetap harus membandingkan jadwal pelajaran yang satu dengan yang lainnya. Sebelumnya Kurniawan [4] telah melakukan penelitian yang membahas mengenai aplikasi jadwal pelajaran ini. Pada aplikasinya telah dilengkapi dengan sebuah fitur yang dapat memberi peringatan apabila terdapat jadwal yang bentrok sehingga lebih memudahkan pengguna untuk mengetahui jadwal mana saja yang bentrok. Namun proses penyusunan jadwal pelajaran tersebut masih tetap dilakukan oleh pengguna sehingga dalam proses pembuatannya masih tetap memakan waktu yang relatif lama agar menghasilkan jadwal pelajaran yang tak memiliki bentrok. Maka penelitian ini difokuskan untuk mengembangkan aplikasi tersebut dengan menambahkan fitur untuk menyusun jadwal pelajaran dan mencegah bentrok antar jadwal pelajaran secara otomatis yang diharapkan dapat membantu pengguna dalam menyusun jadwal pelajaran dengan lebih mudah dan lebih cepat.

II. TINJAUAN PUSTAKA

Jadwal adalah pembagian waktu berdasarkan rencana pengaturan urutan kerja, daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci.

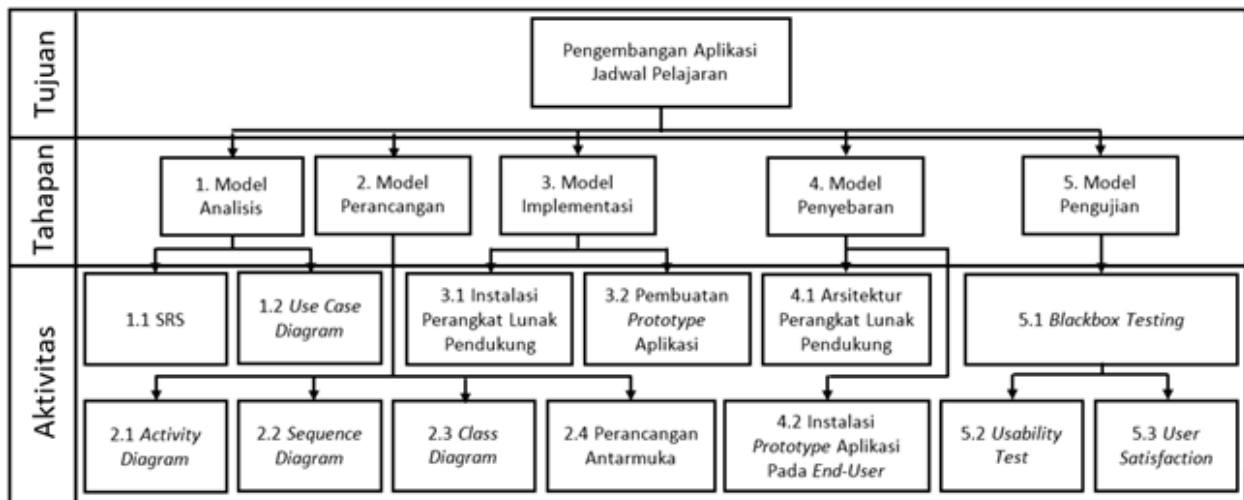
Sedangkan pengertian penjadwalan adalah proses, cara, perbuatan menjadwalkan atau memasukkan ke dalam jadwal [3]. Jadwal didefinisikan sebagai sesuatu yang menjelaskan di mana dan kapan orang-orang dan sumber daya berada pada suatu waktu [1].

Metode yang digunakan dalam pembuatan aplikasi ini adalah metode USDP (*Unified Software Development Process*). USDP sendiri merupakan salah satu metode yang digunakan untuk mengembangkan rekayasa perangkat lunak berorientasi objek yang secara konsisten beradaptasi dengan semakin besar serta semakin kompleksnya sistem-sistem / perangkat lunak yang dikembangkan oleh vendor perangkat lunak seluruh dunia [1]. Metode pengembangan aplikasi perangkat lunak ini terdiri dari 5 tahapan yaitu model analisis, model perancangan, model implementasi, model penyebaran, model pengujian [5].

Pada tahap model analisis dan model perancangan, digunakan pemodelan UML (*Unified Modeling Language*). UML adalah perkakas / *tool* untuk analisis dan perancangan [1]. Sementara itu, untuk pengujian digunakan metode *black box* atau *Black box testing* (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program [6]. Metode *black box* ini terdiri dari *usability test* dan *user satisfaction*, yang akan mengidentifikasi kesalahan dalam beberapa kategori, yang mana di antaranya yaitu kesalahan *interface*, fungsi-fungsi yang salah atau hilang, kesalahan pada struktur data ataupun akses *database* eksternal, kesalahan inisialisasi serta terminasi dan lainnya.

III. KERANGKA KERJA KONSEPTUAL

Berdasarkan tujuan dan literatur yang telah dibahas sebelumnya, WBS (*Work Breakdown Structure*) yang dirancang dengan mengikuti tahapan pada metode USDP dapat digambarkan seperti pada gambar di bawah ini :



Gambar 1 : WBS Pengembangan Aplikasi Penyusun Jadwal Pelajaran Secara Otomatis

Pada tahap model analisis, terdapat aktivitas penentuan SRS dan *use case diagram* yang bertujuan untuk mengetahui kebutuhan apa saja yang diperlukan dalam pengembangan aplikasi penyusun jadwal secara otomatis ini serta penentuan aktor / *user* yang akan menggunakan aplikasi tersebut. Selanjutnya pengumpulan data dilakukan dengan mewawancarai staf pembuat aplikasi dari tiga sekolah yang berbeda yakni SMA (Sekolah Menengah Atas), Sekolah Menengah Kejuruan (SMK), dan MA (Madrasah Aliyyah) sebagai bahan penilaian dari tanggapan *user* mengenai fitur-fitur yang dibutuhkan dalam aplikasi.

Tahap model perancangan, merupakan tahap pemodelan dengan menggunakan UML yang terdiri dari *use case diagram*, *activity diagram*, *sequence diagram* dan *class diagram* serta pembuatan desain tampilan aplikasi dengan menggunakan Microsoft visio yang dapat di *download* pada halaman *website* <http://www.microsoft-visio.com>.

Selanjutnya dilakukan tahapan implementasi yang terdiri dari aktifitas instalasi perangkat-perangkat lunak pendukung serta tahap pembuatan program aplikasi dengan menggunakan Windows Presentation Foundation yang terdapat dalam paket aplikasi Visual Studio yang dapat di *download* pada halaman *website* <https://www.visualstudio.com> serta untuk mengelola basis data sebagai penyimpanan data dari aplikasi tersebut menggunakan phpMyAdmin yang terdapat dalam paket aplikasi XAMPP yang dapat di *download* pada halaman *website* <https://www.apachefriends.org>.

Tahapan selanjutnya ialah model penyebaran, dalam tahap ini dilakukan distribusi aplikasi terhadap pengguna akhir (*enduser*) yaitu staf pembuat jadwal di tiga sekolah yang berbeda yakni SMA, SMK dan MA.

Untuk selanjutnya melakukan pengujian aplikasi dengan menggunakan metode *Black box*. Dalam pengujian tersebut, terdapat dua pengujian yang dilakukan yaitu *usability test* dan *user satisfaction*. Untuk *usability test*, dilakukan instalasi aplikasi pada komputer *desktop* dan staf pembuat jadwal diminta untuk mencoba aplikasi dan melakukan beberapa *task* untuk menguji fungsi-fungsi pada aplikasi penyusun jadwal pelajaran secara otomatis ini. Sementara pada *user satisfaction*, staf pembuat jadwal diberikan kuesioner berupa beberapa pertanyaan yang diajukan sebagai bahan penilaian mengenai kemudahan dalam penggunaan aplikasi, *user interface* dan sebagainya.

VI. HASIL DAN PEMBAHASAN

A. Pencapaian Tujuan

Aplikasi penyusun jadwal pelajaran secara otomatis adalah aplikasi yang dapat menyusun jadwal pelajaran setingkat SMA/Sederajat secara otomatis tanpa adanya jadwal pelajaran yang bentrok. Dengan merujuk pada permasalahan yang telah teridentifikasi yaitu tidak adanya fitur untuk menyusun jadwal pelajaran secara otomatis dan mencegah bentrok antar jadwal pelajaran SMA/Sederajat tersebut. Penelitian ini berhasil mencapai tujuan yaitu mengembangkan aplikasi yang dapat menyusun jadwal secara otomatis dan juga dapat mencegah bentrok antar jadwal pelajaran ketika dalam proses pembuatannya.

B. Penyelesaian Masalah

Adapun dalam proses tercapainya tujuan penelitian diperoleh dari tahapan Metode USDP serta mengikuti susunan kerangka kerja konseptual yang dapat dilihat pada tabel 4.1. Tabel rincian hasil tahapan tersebut telah disesuaikan dengan teknis pembuatan model pada metode UML yang dapat dilihat pada tabel di bawah ini :

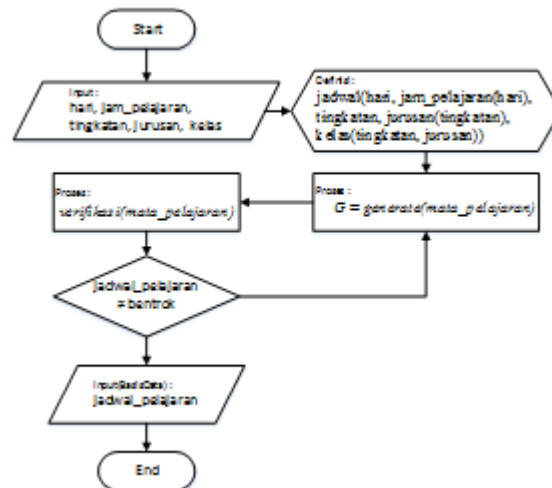
Tabel 1 : Hasil Tahapan Pengembangan Aplikasi

Tahapan Pengembangan USDP	Hasil
1. Model Analisis	1. <i>System Requirement Spesification</i> 2. <i>Use Case</i> a. <i>Use case diagram</i> buat jadwal b. <i>Use case diagram</i> ubah jadwal c. <i>Use case diagram</i> hapus jadwal
2. Model Perancangan	1. <i>Activity Diagram</i> a. <i>Activity diagram</i> buat jadwal b. <i>Activity diagram</i> ubah jadwal c. <i>Activity diagram</i> hapus jadwal 2. <i>Sequence Diagram</i> a. <i>Sequence diagram</i> buat jadwal b. <i>Sequence diagram</i> ubah jadwal c. <i>Sequence diagram</i> hapus jadwal 3. <i>Class Diagram</i> 4. Perancangan Antarmuka

Tabel 1 : Hasil Tahapan Pengembangan Aplikasi Lanjutan

Tahapan Pengembangan USDP	Hasil
3. Model Implementasi	1. Instalasi Perangkat Lunak Pendukung 2. Pembuatan <i>Prototype</i> Aplikasi
4. Model Penyebaran	1. Arsitektur Perangkat Lunak Pendukung 2. Instalasi <i>Prototype</i> Aplikasi Pada <i>User</i>
5. Model Pengujian	1. <i>Usability Test</i> 2. <i>User Satisfaction</i>

Pada tahap akan dijelaskan proses dari penyusunan jadwal tersebut yang dilakukan oleh sistem sebagai berikut :

Gambar 2 : *Flowchar* Proses Penyusunan Jadwal Pelajaran Oleh Sistem

Sistem meminta nilai input seperti yang tertera pada gambar yaitu hari, jam_pelajaran, angkatan, jurusan, kelas, mata pelajaran, dan bobot pelajaran. Tahap selanjutnya sistem akan melakukan *generate(mata_pelajaran)* yaitu suatu proses di mana mata pelajaran akan dipilih secara acak (*random*) oleh sistem. Untuk menghindari terjadinya bentrok dari hasil pengacakan jadwal pelajaran, maka dilakukan *verifikasi(jadwal_pelajaran)* yaitu dengan cara membandingkan satu baris pada hari yang sama di tiap-tiap kelas yang berbeda. Bila masih terdapat jadwal yang bentrok antar kelas maka sistem akan melakukan lagi *generate* dan *verifikasi* hingga tidak ada lagi jadwal yang bentrok. Bila tidak ada lagi jadwal yang bentrok maka jadwal tersebut selanjutnya akan dimasukkan ke dalam basis data yang telah disediakan agar dapat ditampilkan kembali untuk selanjutnya. Agar lebih jelas, akan diterangkan proses penyusunan jadwal dengan sebuah contoh sebagai berikut :

1. Input :

hari = 6

//Maksudnya jumlah hari belajar-mengajar dalam 1 minggu. Contoh hari= 0 (senin), //hari =1 (selasa), hari = 2 (rabu), hari=3 (kamis), hari = 4 (jum'at), hari = 5 (sabtu).

jam_pelajaran(hari) = jp(hari)

//Maksudnya adalah jam pelajaran pada hari yang berbeda bisa memiliki jumlah

//jam pelajaran yang berbeda. Contoh jam_pelajaran(0) =8, jam_pelajaran(1)= 8,

//jam_pelajaran(2)=8, jam_pelajaran(3)=8, jam_pelajaran(4)=5, jam_pelajaran(5)=8

tingkatan = 3

//maksudnya adalah jumlah tingkatan dalam satu sekolah. Contoh tingkat=0 (X),

//tingkat=1(XI) dan tingkat= 3(XII).

jurusan(tingkatan) = j(tingkatan)

```

//maksudnya adalah jurusan pada tiap angkatan yang berbeda bisa memiliki jumlah
//jurusan yang berbeda. Contoh jurusan(0)= 0, jurusan(1) = 2, jurusan(2) = 2.
kelas(tingkatan,jurusan) = k(tingkatan,jurusan)
//maksudnya adalah tiap kelas pada tingkatan yang berbeda & jurusan yang berbeda
//bisa memiliki jumlah kelas yang berbeda pula. Contoh kelas(0,0) = 10,
//kelas(1,0) = 4, kelas(1,1) = 3.
mata_pelajaran = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
22, 23, 24, 25, 26, 27, 28, 29}
//Contoh mata_pelajaran=0 (matematika), mata_pelajaran=1 (bahasa Indonesia) dst.
bobot(mata_pelajaran) = b(mata_pelajaran)
//Maksudnya adalah bobot tiap mata pelajaran yang berbeda bisa memiliki bobot
//yang berbeda. Contoh bobot(0) = 2 (2 jam pelajaran/minggu), bobot(1) = 1 (1 jam
//pelajaran/minggu), bobot(2) = 5 (5 jam pelajaran/minggu) dan begitu seterusnya
//hingga semua nilai bobot mata pelajaran terisi.

```

2. Definisi

```

jadwal(hari, jam_pelajaran(hari), tingkatan, jurusan(tingkatan), kelas(tingkatan, jurusan))
//jadwal merupakan array lima dimensi yang tersusun atas hari, jam pelajaran,
//tingkatan, jurusan, dan kelas.

```

3. *generate(mata_pelajaran)*

Setelah didapat nilai pada *variable-variable* sebelumnya, maka proses selanjutnya adalah *generate(mata_pelajaran)*. *mata_pelajaran* merupakan sebuah *variable* yang berisi kode dari mata pelajaran yang tersedia pada sekolah tertentu. Hasil dari *generate(mata_pelajaran)* dimasukkan ke dalam *variable* G. Agar dapat memahami proses dari *generate(mata_pelajaran)* akan diterangkan dalam pembahasan sebagai berikut :

```

G = generate(mata_pelajaran)

```

```

//Melakukan pemilihan secara acak (random) dari kode mata pelajaran yang ada
//yaitu dari 1-30. Contoh G = 25, G = 18, G = 15 dst. hingga semua jadwal terisi.

```

4. *verifikasi(mata_pelajaran)*

Kode mata pelajaran yang telah dimasukkan ke dalam *variable* G selanjutnya akan melalui tahap *verifikasi(mata_pelajaran)*. Pada tahap ini akan dilakukan pemeriksaan terhadap jadwal sekarang (G) dengan jadwal-jadwal yang lain (jadwal(h, jp(h), t, j(t), k(t, j))) pada baris (jam pelajaran yang sama). Berikut proses pemeriksaannya :

```

For t = 0 to t = tingkatan do
  For j = 0 to j = jurusan(tingkatan) do
    For k = 0 to k = kelas(tingkatan, jurusan(tingkatan))
      If G ≠ jadwal(t,j,k) then
        jadwal(c_h, c_jp(h), c_t, c_j(t), c_k(t, j)) = G
      Else
        Goto Generate(mata_pelajaran)
      End If
    End For
  End For
End For

```

Selama proses perulangan belum habis ($t \neq$ tingkatan) maka proses pemeriksaan akan terus dilakukan walaupun *variable* G telah memenuhi persyaratan dan di masukkan ke *variable* jadwal(c_h, c_jp(h), c_t, c_j(t), c_k(t, j)). Nilai *variable* G akan direset lagi sampai ditemukan nilai mata pelajaran yang benar-benar tidak sama dengan *variable* jadwal(t,j,k).

5. Menyimpan ke dalam basis data

Tahap terakhir adalah menyimpannya ke dalam basis data. Hal ini dilakukan agar ketika aplikasi jadwal ditutup dan dibuka kembali aplikasi jadwal tidak melakukan *generate* ulang, selain karena akan memakan waktu yang relatif lama juga jadwal yang dihasilkan pada tiap-tiap *generate* akan selalu berbeda. Maka dari itu jadwal pelajaran dan seluruh

variable yang terkait dengan jadwal pelajaran akan dimasukkan ke basis data.

Gambar 3 : Implementasi Antarmuka Penyusun Jadwal Pelajaran Otomatis

V. KESIMPULAN

Berdasarkan pada hasil penelitian yang telah dilakukan sebelumnya dan uraian yang telah dijelaskan pada pembahasan, dengan demikian dapat disimpulkan bahwa :

1. Proses penelitian tugas akhir ini berhasil mencapai tujuan yakni, mengembangkan aplikasi penyusun jadwal pelajaran secara otomatis yang memiliki fitur dapat menyusun dan mencegah bentrok jadwal pelajaran yang didasarkan pada rancangan yang telah dilakukan.
2. Aplikasi yang telah dibangun dengan fitur penyusun jadwal secara otomatis dan pencegah bentrok antar jadwal di dalamnya menjadi solusi bagi penyelesaian masalah seputar penyusunan jadwal pelajaran yakni relatif lamanya penyusunan jadwal pelajaran dan masih sering terjadinya bentrok antar jadwal pelajaran yang telah tersusun.
3. Sesuai dengan hasil pengujian berdasarkan pada *user satisfaction* di dapat persentase sebagai berikut untuk kepuasan terhadap aplikasi penyusunan jadwal pelajaran secara otomatis 73% Sangat baik 27% Baik, 0% kurang baik, 0% buruk, maka dapat disimpulkan bahwa aplikasi penyusun jadwal pelajaran secara otomatis ini dapat membantu dalam melakukan penyusunan jadwal pelajaran.

UCAPAN TERIMAKASIH

Penulis E.H mengucapkan banyak terima kasih kepada kedua orang tua penyusun yang telah banyak memberi dukungan baik secara moril maupun materiil. Penulis juga mengucapkan banyak terima kasih kepada Bapak Rinda Cahyana, MT selaku pembimbing yang telah memberikan arahan serta bimbingan selama penyelesaian laporan penelitian ini

DAFTAR PUSTAKA

- [1] Buyens, J. (2001). *Web Database Development*. Jakarta : Elex Media Komputindo
- [2] Chambers, R. 1995. "Poverty and Livelihood: Whose Reality Count?" Dalam: *People From Improverishment to Empowemnet*. New York: Uner Kirdar dan Leonard Silk (Eds), New York University Press.
- [3] Depdikbud (1995). *Kamus Besar Bahasa Indonesia*. Jakarta : Balai Pustaka
- [4] Kurniawan, F. (2013). *Sistem Penyusunan Jadwal Pelajaran Berbasis Web*. Universitas Negeri Yogyakarta
- [5] Nugroho, A. (2010). *Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP*. Yogyakarta : Andi OFFSET
- [6] Rosa, A.S & Shalahuddin, M. (2011). *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung : Modula