

# Perancangan Aplikasi Pelacakan Program Kerja Kampus di STMIK Antar Bangsa

Subhiyanto<sup>1</sup>, Aristejo<sup>2</sup>, Esthi Adityarini<sup>3</sup>

**Abstract**— Every agency or company certainly has a work program for the next one year, this is also carried out at STMIK Antar Bangsa. However, with the passage of time, often during the evaluation, problems of incompatibility were found between the previously designed work programs and their implementation. To overcome this problem we propose a work program tracking application design (APELPROKER) using the waterfall method with the User Acceptance Test (UAT), testing technique which produces 93.75% of respondent responses with the aim of being able to help solve criteria problems very well.

**Intisari**— Setiap instansi atau perusahaan tentu memiliki program kerja selama satu tahun kedepan, hal ini pun dilakukan pada STMIK Antar Bangsa. Namun berjalannya waktu seringkali pada saat evaluasi sering ditemukan masalah ketidaksesuaian antara program kerja yang sebelumnya dirancang dengan pelaksanaannya. Untuk mengatasi masalah tersebut kami mengusulkan rancangan aplikasi pelacakan program kerja (APELPROKER) menggunakan metode waterfall dengan teknik pengujian UAT yang menghasilkan tanggapan responden sebanyak 93,75% dengan tujuan dapat membantu menyelesaikan masalah kriteria sangat baik.

**Kata Kunci**— Aplikasi, Prototype, Waterfall, Program Kerja.

## I. PENDAHULUAN

Seluruh instansi, khususnya kampus tentu akan membuat program kerja untuk mencapai visinya. Program kerja akan dibuat setiap awal tahun, kemudian dilaksanakan dan dilakukan evaluasi pada akhir tahun melalui laporan-laporannya.

Pada saat evaluasi, seringkali muncul masalah bahwa ternyata program yang dilaksanakan tidak sesuai dengan program yang direncanakan. Ketidaksesuaiannya ini bersifat mikro, artinya penyimpangannya tidak jauh dari tujuan program yang direncanakan tetapi tidak dapat memenuhi kriteria evaluasi program kerja tersebut.

Tak terkecuali pada STMIK Antar Bangsa, salah satu sekolah tinggi informatika dan komputer yang terdapat di Tangerang, Banten. Dalam rangka mewujudkan cita-cita organisasi yang baik, maka perlu dibuatkan sebuah program kerja untuk dijadikan pegangan bagi organisasi.

<sup>1,2</sup> STMIK Antar Bangsa, Kawasan Bisnis CBD Ciledug, Jl. HOS Cokroaminoto No.29-35, RT.001/RW.001, Karang Tengah, Kec. Ciledug, Kota Tangerang, Banten 15157 (telp: 0811-9391-441 e-mail: [subhiyanto.bian@gmail.com](mailto:subhiyanto.bian@gmail.com), [aristejo14@gmail.com](mailto:aristejo14@gmail.com)).

<sup>3</sup> Institut Daarul Qur'an Jakarta, Kawasan Bisnis CBD Ciledug, Jl. Cipondoh Makmur Raya, RT.003/RW.009, Cipondoh Makmur, Kec. Cipondoh, Kota Tangerang, Banten 15148 (telp: 021-3111-8118 e-mail: [esthi.aditya@gmail.com](mailto:esthi.aditya@gmail.com)).

Penyebab masalah ini muncul adalah pada saat pelaksanaan seringkali terjadi ketidaksesuaian antara pelaksanaan program dengan program kerja yang telah ditentukan di awal tahun. Rata-rata alasannya karena terjadi penyesuaian lapangan pada saat program kerja tersebut dilaksanakan.

Faktor penyesuaian yang dimaksud bisa bermacam-macam mulai dari tidak tersedianya sumber daya yang ditentukan dalam program, kekurangan biaya, kondisi lapangan yang berubah ketika program dilaksanakan, berubahnya asumsi yang digunakan pada saat program dibuat sehingga pada saat pelaksanaan, asumsi tersebut sudah tidak dapat digunakan lagi, dan lainnya.

Jika hal ini terus berlangsung maka pada satu titik akan menjadi ganjalan bagi kampus untuk mencapai visinya. Misal menjadi ganjalan dalam proses evaluasi, akreditasi, dan sebagainya. Untuk memecahkan penyebab kedua ini maka sebuah kampus perlu untuk menggunakan aplikasi pelacakan program kerja (APELPROKER).

APELPROKER yang akan dibuat adalah dalam bentuk perancangan aplikasi atau biasa disebut *prototype*. *Prototype* adalah sebuah gambaran yang memberikan ide bagi para pengembang dan calon penngguna, bagaimana system akan berfungsi dan berjalan [1].

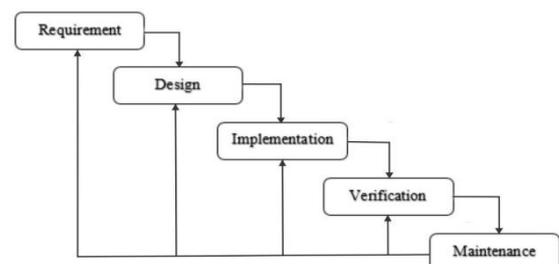
## II. METODE PENELITIAN

### A. Metode Waterfall

Metode air terjun atau yang sering disebut metode waterfall seing dinamakan siklus hidup klasik (*classic life cycle*), nama model ini sebenarnya adalah "*Linear Sequential Model*" dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modelling*), konstruksi (*contruction*), serta penyerahan sistem ke para pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan [2].

### B. Tahapan Metode Waterfall

Tahapan dari metode *waterfall* dapat dilihat dalam gambar 1 berikut ini :



Gambar 1. Tahapan Metode Waterfall [3]

Metode *Waterfall* terbagi menjadi lima tahap yang terstruktur, yaitu sebagai berikut [2]:

1. *Requirement*

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

2. *Design*

Pada tahap ini, pengembang membuat desain sistem yang dapat membantu menentukan perangkat keras (hardware) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3. *Implementation*

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit testing.

4. *Verification*

Pada tahap ini, sistem dilakukan verifikasi dan pengujian apakah sistem sepenuhnya atau sebagian memenuhi persyaratan sistem, pengujian dapat dikategorikan ke dalam unit testing (dilakukan pada modul tertentu kode), sistem pengujian (untuk melihat bagaimana system bereaksi ketika semua modul yang terintegrasi) dan penerimaan pengujian (dilakukan dengan atau nama pelanggan untuk melihat apakah semua kebutuhan pelanggan puas).

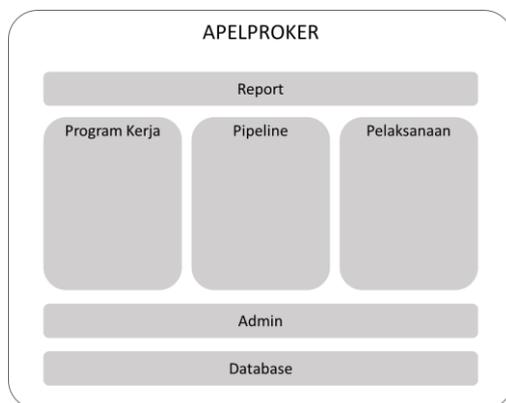
5. *Maintenance*

Ini adalah tahap akhir dari metode waterfall. Perangkat lunak yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

III. PEMBAHASAN DAN HASIL

A. *Requirement*

Konsep dasar APELPROKER adalah mencatat program, mencatat *pipeline*, dan mencatat hasil. Dengan konsep ini maka APELPROKER memiliki tiga komponen yang saling terhubung sehingga akan membuatnya dapat melakukan pengecekan apakah komponen-komponen dalam program kerja sudah terlaksana. Hasil pengecekan akan ditampilkan dalam bentuk laporan. Berikut adalah rancangan kebutuhan dari APELPROKER di STMIK Antar Bangsa:



Gambar 2. Rancangan Kebutuhan

Pengguna APELPROKER adalah manajemen madya, manajemen atas, dan admin kampus. Tentunya semua pihak akan menggunakan sesuai fungsinya masing-masing dan ini diatur dengan menggunakan hak akses yang terdapat di lapisan Admin. Dengan melihat pengguna APELPROKER maka aplikasi ini dirancang dengan *platform web base* sehingga memudahkan para pengguna untuk mengakses dari lokasi manapun ketika sedang berada diluar kampus.

Di dalam APELPROKER rancangan modul yang akan dibangun adalah seperti berikut:

1. Program Kerja

Modul program kerja berisi program kerja yang akan dilakukan beserta detilnya. Detil merupakan komponen dari program kerja. Pengguna dapat menambahkan program kerja sebanyak yang dibutuhkan. Jika dilakukan penambahan program kerja maka setiap program kerja yang dibuat dapat dibuat pula detilnya. Detil program kerja sifatnya bebas sesuai dengan kebutuhan program kerja yang dibuat. Artinya detil antara satu program kerja dengan lainnya dapat berbeda, baik jumlah maupun namanya.

2. *Pipeline*

*Pipeline* adalah tahapan yang sedang berjalan dari sebuah program kerja. Pada APELPROKER ini, *Pipeline* yang disediakan adalah Mulai, Sedang Berjalan, dan Selesai. *Pipeline* dikenakan kepada setiap detil program kerja. Tujuannya adalah supaya tidak ada lagi detil yang terlewat dikerjakan pada akhir masa program kerja. Pada modul *Pipeline* ini juga terdapat tanggal mulai dan Tanggal target selesai. Fungsi tanggal ini memberikan poin perhatian kepada pengguna jika detil program kerja belum selesai namun telah melewati tanggal target selesai. Pada *pipeline* ini terdapat fitur untuk perubahan status menjadi selesai atau terlambat.

3. Pelaksanaan

Modul pelaksanaan merupakan recap dari setiap detil program kerja yang telah diselesaikan. Di sini akan terdapat daftar program kerja yang detilnya telah berubah statusnya menjadi selesai. Salah satu detil program kerja statusnya telah selesai maka akan masuk ke modul Pelaksanaan ini. Dengan demikian maka pengguna akan mudah untuk mengontrol pelaksanaan aktivitas detil pada setiap program kerja yang dibuat.

4. Admin

Modul admin ini berisi semua parameter yang digunakan sebagai setting APELPROKER seperti manajemen pengguna. Bagian Manajemen Pengguna membantu user admin untuk mendaftarkan, memberi hak akses, dan menghapus pengguna APELPROKER sesuai dengan kebutuhannya.

5. *Report*

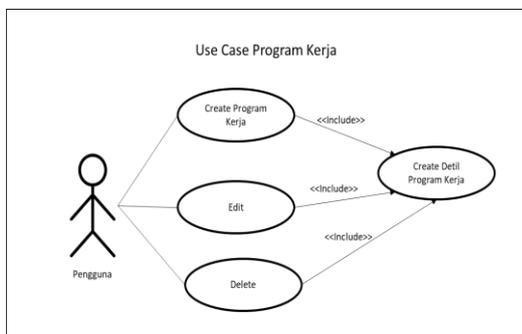
Fungsi modul *report* adalah menampilkan laporan-laporan terkait pelacakan program kerja. Laporan-laporan ini sifatnya *pre define*, artinya laporan-laporan ini sudah disiapkan sebelumnya saat proses development. Di sini pengguna tinggal memilih laporan yang ingin ditampilkan dengan memasukkan parameter-parameter yang tersedia seperti tanggal awal dan tanggal akhir laporan, dan beberapa parameter lainnya.

B. Design

Pada tahap ini yang dilakukan adalah proses perancangan untuk APELPROKER yang meliputi *use case* program kerja dan *usecase report*, yang dapat digambarkan sebagai berikut:

1. Use Case Program Kerja

Use case diagram adalah pemodelan untuk menggambarkan behavior sistem yang akan dibuat. Diagram use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang dapat di buat [4]. Pada use case Program Kerja, pengguna dapat melakukan Create Program Kerja, Edit Edit dan Delete. Ketika selesai membuat program kerja maka system akan menampilkan layar Deil Program kerja untuk diisi datanya oleh Pengguna. Ketika akan melakukan Edit, maka user dapat mengedit Program Kerjanya saja, Detilnya saja, atau keduanya. Ketika akan melakukan Delete maka user dapat menghapus detilnya atau menghapus dari program kerjanya. Jika user menghapus dari program kerjanya maka seluruh data detil program kerja akan ikut terhapus.



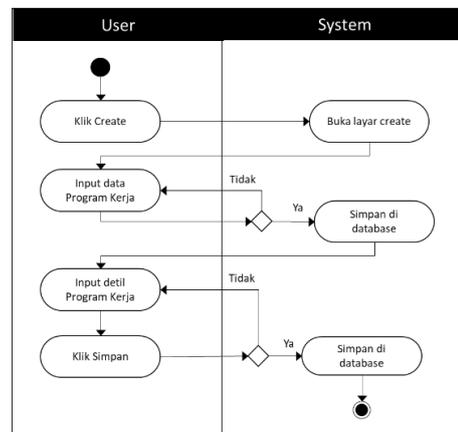
Gambar 3. Use Case Program Kerja

Penjelasan lebih detil dari use case Create, Edit, dan Delete dituliskan dalam tabel berikut :

Tabel 1. Tabel Use Case Create

Use Case Create	
<b>Aktor</b>	Pengguna yang sudah didaftarkan
<b>Trigger</b>	Klik tombol create
<b>Pre-Conditions</b>	Tidak ada
<b>Post-Conditions</b>	Terbentuk data Program Kerja. Jika data program kerja sudah disimpan maka APELPROKER akan menampilkan layar detil Program Kerja. Jika user sudah selesai memasukkan data Detil Program Kerja maka user dapat menyimpannya ke dalam database.
<b>Workflow Use Case</b>	Tercantum dalam gambar Activity Create
<b>Kondisi Alternatif</b>	Jika terjadi error dari system maupun error dari pengguna maka system akan menampilkan pesan error dan menahan proses selanjutnya.
<b>Kebutuhan Khusus</b>	Tidak ada

Activity diagram adalah suatu diagram yang menggambarkan konsep aliran data/kontrol, aksi terstruktur serta dirancang dengan baik dalam suatu system [5]. Berikut ini adalah activity diagram untuk create,:



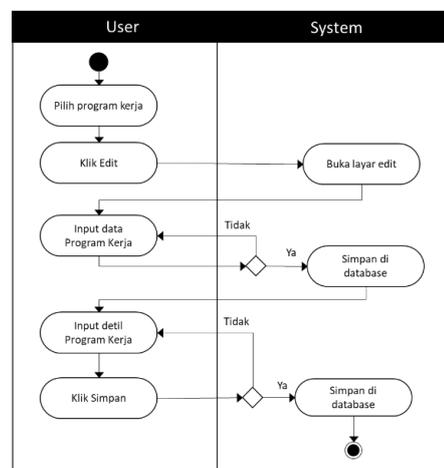
Gambar 4. Activity Diagram Create

Berikut ini adalah penjelasan mengenai use case edit:

Tabel 2. Tabel Use Case Edit

Use Case Edit	
<b>Aktor</b>	Pengguna yang sudah didaftarkan
<b>Trigger</b>	Klik tombol edit
<b>Pre-Conditions</b>	Program kerja yang akan di edit telah dipilih terlebih dahulu
<b>Post-Conditions</b>	Data tersimpan dalam database
<b>Workflow Use Case</b>	Tercantum dalam gambar activity edit
<b>Kondisi Alternatif</b>	Jika terjadi error dari system maupun error dari pengguna maka system akan menampilkan pesan error dan menahan proses selanjutnya.
<b>Kebutuhan Khusus</b>	Tidak ada

Berikut ini adalah activity diagram dari edit :



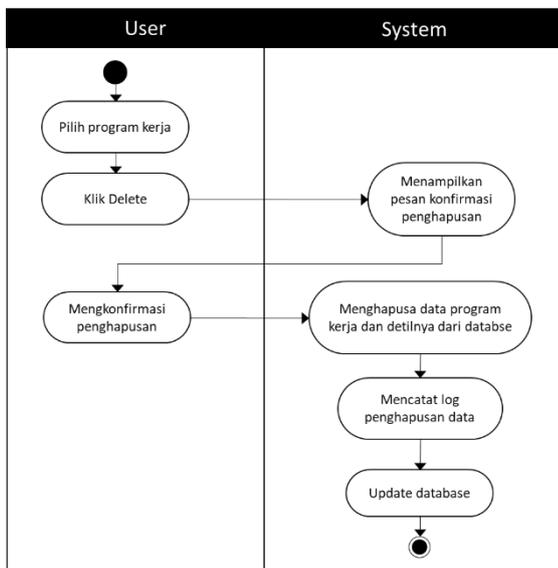
Gambar 5. Activity Diagram Edit

Berikut ini adalah penjelasan mengenai use case *delete*:

Tabel 3. Tabel *Use Case Delete*

Use Case Delete	
<b>Aktor</b>	Pengguna yang sudah didaftarkan
<b>Trigger</b>	Klik tombol Delete
<b>Pre-Conditions</b>	Program kerja yang akan di delete telah dipilih terlebih dahulu
<b>Post-Conditions</b>	Data program kerja dan detilnya dihapus dari database. System mencatat log penghapusan data.
<b>Workflow Use Case</b>	Tercantum dalam gambar Activity Delete
<b>Kondisi Alternatif</b>	Jika terjadi error dari system maupun error dari pengguna maka system akan menampilkan pesan error dan menahan proses selanjutnya.
<b>Kebutuhan Khusus</b>	Tidak ada

Berikut ini adalah activity diagram dari *delete* :

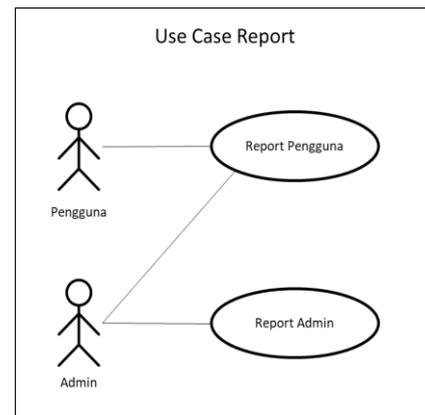


Gambar 6. *Activity Diagram Delete*

## 2. Use Case Report

User modul Report adalah pengguna dan admin. Report pada APELPROKER juga ada dua jenis yaitu report untuk pengguna dan report untuk admin. Sistem menampilkan report melalui dua cara:

- Berdasarkan parameter yang telah diisi oleh user.
- Berdasarkan paramtere default yang di set di dalam system. Jika user mengosongkan parameter maka system menampilkan laporan dengan cara ini.



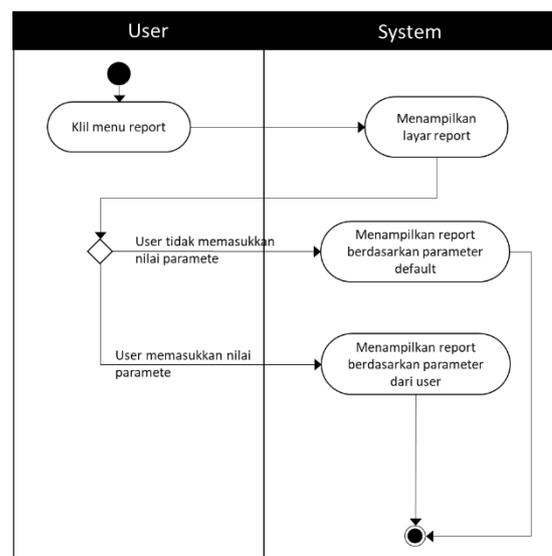
Gambar 7. *Use Case Report*

Berikut ini adalah penjelasan mengenai use case *report*:

Tabel 4. Tabel *Use Case Report*

Use Case Delete	
<b>Aktor</b>	Pengguna dan Admin yang sudah didaftarkan
<b>Trigger</b>	Klik tombol Report
<b>Pre-Conditions</b>	Data program kerja dan detilnya telah disimpan di database
<b>Post-Conditions</b>	System menampilkan report berdasarkan data parameter yang dimasukkan atau berdasarkan parameter default.
<b>Workflow Use Case</b>	Tercantum dalam gambar Activity Report
<b>Kondisi Alternatif</b>	Jika terjadi error dari system maupun error dari pengguna maka system akan menampilkan pesan error dan menahan proses selanjutnya.
<b>Kebutuhan Khusus</b>	Tidak ada

Berikut ini adalah activity diagram dari *report* :



Gambar 7. *Activity Diagram Report*

### C. Implementation

*User Interface* adalah bagian visual desain yang mewakili dan mewakili system yang dapat dilihat oleh pengguna [6]. Pada tahap ini, implementasi dilakukan dengan menampilkan *User Interface* APELPROKER yang meliputi *User Interface* program kerja dan laporan, sebagai berikut :

#### 1. *User Interface* Program Kerja

Gambar 8. *User Interface* Program Kerja (Create)

Pada UI ini, user membuat program kerja dan mengisikan detailnya. Contoh detail yang dapat diisikan pada program kerja misalnya dana, kegiatan, dan lainnya. Berikut adalah penjelasan field dan tombolnya.

##### a. Program

- Nama program: diisikan nama program yang akan dibuat.
- Tahap: informasi tentang pipeline program. Saat baru membuat maka otomatis terisi dibuat. Jika sudah mulai berjalan, maka klik tombol Selanjutnya pada interface Edit. Pada Interface Create, tombol selanjutnya masih disable.
- Tanggal: informasi tanggal dibuatnya program. Ini otomatis terisi oleh system. Jika anda melakukan perubahan pada nama program maka tanggal akan otomatis terganti menjadi tanggal terakhir dilakukan perubahan.
- Tombol Submit: Menyimpan nama program dan detailnya yang telah dibuat.
- Cancel: membatalkannya penyimpanan program dan detailnya. Interface akan menampilkan konfirmasi dan mengosongkan seluruh field.

##### b. Detail Program

Bagian detail program jumlahnya tidak dibatasi. Anda dapat menambahkan detail sesuai kebutuhan.

- Label: Isi dengan nama yang anda perlukan. Misal: Biaya.
- Type: tipe ini berisi tipe dari field nilai yang akan anda isikan. Misal anda membuat detail Biaya, maka pada field label, anda pilih Numeric sehingga anda dapat mengisikan nominal biaya pada field nilai.
- Type: berisi pilihan properties field Nilai yang akan diisi. Piliannya adalah: numeric untuk angka, alphabet untuk teks, date untuk tanggal.

- Nilai: isi dengan nilai untuk label yang anda buat sesuai dengan field type yang telah anda pilih sebelumnya.
- Tanggal: berisi tanggal saat anda membuat sebuah detail. Field ini otomatis terisi oleh system.
- Tombol Add Detail: klik untuk menambahkan detail selanjutnya.
- Tombol Delete Detail: klik untuk menghapus Detail

Saat user klik edit program kerja maka yang muncul pertama adalah interface berikut:

Gambar 9. *User Interface* Program Kerja (Edit)

Disini user memilih program kerja mana yang akan di edit atau di delete. Setelah memilih, maka checkbox akan terisi dan user dapat memilih edit atau delete. Jika user mengklik delete maka data program kerja yang dipilih akan dihapus dari system beserta seluruh data detailnya. Jika user klik edit, maka akan tampil interface berikut:

Gambar 10. *User Interface* Program Kerja (Edit dan Delete)

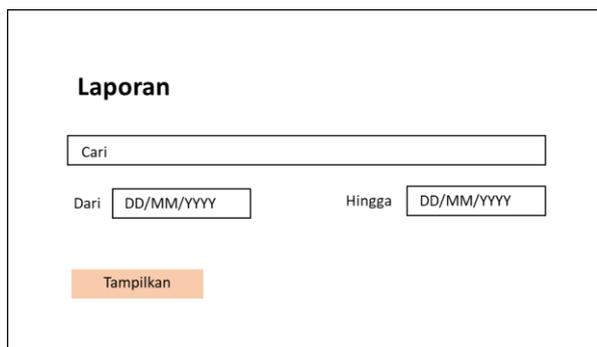
*User Interface* edit sama dengan *User Interface* create. Perbedaannya hanya di *User Interface* edit, field-field nya sudah terisi dengan data yang telah diisikan sebelumnya dan anda dapat mengubah isi field-field tersebut. Pada *User Interface* edit, tombol selanjutnya akan aktif. Tujuan tombol ini adalah mengubah setatus tahap. Ketika anda klik maka status tahap akan berubah dari dibuat menjadi dalam proses dan seterusnya.

Ketika tahap berubah maka tanggal juga akan otomatis berubah. Interface Edit memiliki field Realisasi pada setiap detail program kerja. Field realisasi ini adalah sebagai media untuk memasukkan informasi realisasi yang telah dilakukan untuk

setiap detail. Sehingga pada akhir masa program kerja terlihat antara rencana dan realisasi. Pada interface edit ini juga terdapat tombol attachment untuk melampirkan dokumen pada setiap detail program.

2. *User Interface* Laporan

Laporan berfungsi untuk menampilkan laporan-laporan tentang program kerja termasuk detailnya, tahapannya dan lainnya sesuai dengan yang telah didefinisikan sebelum mulai development APELPROKER. Sehingga akan jelas terlihat bagaimana tahapan program-program kerja yang telah dibuat dan bagaimana hasilnya. Saat mengklik menu laporan, maka akan tampil interface berikut :



Gambar 11. *User Interface* Cari Laporan

Disini user mengisi free text pada field cari. Caranya sama seperti saat kita melakukan pencarian di google. Selanjutnya uaser dapat memasukkan parameter rentang tanggal. System APELPROKER memiliki parameter default yang di hardcode di aplikasi. Sehingga jika user mengosongkan semua field ini maka APELPROKER akan menampilkan laporan berdasarkan parameter default tersebut. Contoh laporan APELPROKER adalah seperti berikut:

LAPORAN	
PROGRAM PELATIHAN IMS POWERPOINT UNTUK MEMBUAT GAMBAR SKEMATIK	
Lokasi:	SMK Tirta Jaya Jababeka Tangerang
Realisasi:	SMK Tirta Jaya Jababeka Tangerang
Anggaran:	Rp 3.000.000
Realisasi:	Rp 3.000.000
Target peserta:	50 orang
Realisasi:	70 orang. Terjadi penambahan peminat saat program berjalan.
Pengajar:	5 orang
Realisasi:	5 orang
Tanggal mulai:	3 Juli 2023
Realisasi:	3 Juli 2023
Tanggal selesai:	5 Juli 2023
Realisasi:	5 Juli 2023
Tanggal Program dibuat:	20 Januari 2023
Tanggal program selesai:	2 Agustus 2023
Status:	Selesai

Gambar 12. *User Interface* Cari Laporan

6. *Verification*

Pada tahap ini dilakukan pengujian terhadap *User Interface* yang telah dibuat, Pengujian dilakukan menggunakan *User*

*Acceptance Test* (UAT). Pengujian ini melibatkan pengguna, untuk mengetahui apakah *User Interface* yang dibuat sudah sesuai dengan kebutuhan di STMIK Antar Bangsa. Use case diagram adalah pemodelan untuk menggambarkan behavior sistem yang akan dibuat. Diagram use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang dapat di buat [7]. Pengujian ini harus memenuhi 4 kategori yaitu fungsionalitas, kinerja, kualitas antarmuka dan kualitas perangkat lunak. Perhitungan dari pengujian UAT dapat menggunakan rumus berikut ini.

$$\%SkorTotal = \frac{skorAktual}{skorIdeal} \times 100 \%$$

Skor aktual merupakan jawaban dari seluruh responden dari kuesioner yang telah dibagikan [13]. Sedangkan skor ideal merupakan skor atau bobot tertinggi dari semua responden yang diasumsikan memilih jawaban dengan skor tertinggi. Pembobotan nilai skor aktual dapat dilihat pada tabel 2.

Tabel 5. Kriteria Tanggapan Responden

% Jumlah Skor	Kriteria
20,00% - 36,00%	Tidak Baik
36,01% - 52,00%	Kurang Baik
52,01% - 68,00%	Cukup
68,01% - 84,00%	Baik
84,01% - 100,00%	Sangat Baik

Hasil tanggapan responden terhadap pengujian UAT, dapat dilihat pada table 3.

Tabel 3. Tanggapan Responden terhadap Pengujian UAT

User Accetance Test	Kriteria Jawaban		Hasil Tiap Aspek
	Pass	Fail	
Fungsionalitas	12	0	100%
Kinerja	11	1	91,67%
Kualitas Antar Muka	12	0	100%
Kualitas Perangkat Lunak	10	2	83,33%
<b>Total</b>	<b>45</b>	<b>3</b>	

Dari hasil penyebaran kuesioner terhadap para responden, maka diperoleh nilai sebagai berikut :

$$\begin{aligned} \%SkorTotal &= \frac{skorAktual}{skorIdeal} \times 100 \% \\ &= (45/48) \times 100\% \\ &= 93,75\% \end{aligned}$$

Dari hasil pengolahan dan tanggapan responden User Acceptance Test maka didapatkan skor total yaitu 93,75%. Berdasarkan hal tersebut maka termasuk dalam kriteria "Sangat Baik" untuk perancangan APELPROKER ini.

#### D. Maintenance

Pada tahap akhir ini merupakan pemeliharaan terhadap system yang telah dibuat. Pemeliharaan tersebut termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

### 3. KESIMPULAN

Dari hasil perancangan, desain, implementasi, pengujian dan pemeliharaan yang menggunakan metode *Waterfall*, dapat disimpulkan bahwa perancangan aplikasi pelacakan program kerja di STMIK Antar Bangsa adalah sangat baik untuk di terapkan yang dibuktikan dengan hasil tanggapan responden sebanyak 93,75%. Karena APELPROKER ini dapat menyelesaikan masalah ketidaksesuaian antara pelaksanaan program dengan program kerja yang telah ditentukan di STMIK Antar Bangsa.

### REFERENSI

- [1] Adityarini, E. (2023). Perancangan Aplikasi Dokumentasi Seminar Para Dosen di STMIK Antar Bangsa. *Jurnal Teknik Informatika*, 9 (1), 8-13.
- [2] Wahid, A. A. (2020). Analisis metode waterfall untuk pengembangan sistem informasi. *J. Ilmu-ilmu Inform. dan Manaj. STMIK*, no. November, 1-5.
- [3] R. S. Pressman, *Rekayasa Perangkat Lunak (Pendekatan Praktisi)*, Yogyakarta: Penerbit ANDI, 2012.
- [4] Rahmawati, F. H., & Adityarini, E. (2021). Sistem Informasi Persediaan Barang pada CV. Anak Teladan. *Jurnal Sistem Informasi*, 10(1), 1-7.
- [5] Arianti, T., Fa'izi, A., Adam, S., & Wulandari, M. (2022). Perancangan Sistem Informasi Perpustakaan Menggunakan Diagram UML (Unified Modelling Language). *Jurnal Ilmiah Komputer Terapan dan Informasi*, 1(1), 19-25.
- [6] LaViola Jr, J. J., Kruijff, E., McMahan, R. P., Bowman, D., & Poupyrev, I. P. (2017). *3D user interfaces: theory and practice*. Addison-Wesley Professional. <https://doi.org/10.51998/jsi.v10i1.351>
- [7] Adityarini, E. (2021). Development of Knowledge management system to Support Knowledge Sharing Among Lecturers: Case Study at STMIK Antar Bangsa.

*Systematics*, 3(3), 324.



Subhiyanto, lahir di Brebes pada tanggal 10 Maret 1984. Tahun 2012 Lulus Sarjana Komputer Jurusan Teknik Informatika di STMIK Nusa Mandiri. Tahun 2020 lulus program Pasca Sarjana Ilmu Komputer dengan konsentrasi Rekayasa Komputasi Terapan di Universitas Budi Luhur. Saat ini aktif mengajar sebagai dosen tetap dan sebagai Kepala Biro Teknologi Informasi di STMIK Antar Bangsa.



Aristejo, lahir di Jakarta pada tanggal 14 Oktober 1978. Tahun 2001 Lulus dari Jurusan Teknik Sipil di Universitas Trisakti. Tahun 2004 lulus program Pasca Sarjana Ilmu Komputer dengan di Universitas Indonesia. Saat ini aktif mengajar sebagai dosen tetap pada program studi Teknik Informatika di STMIK Antar Bangsa.



Esthi Adityarini. Lahir di Jakarta, pada Tanggal 31 Januari 1994. Tahun 2015 lulus dari Program Strata Satu (S1) Jurusan Teknik Informatika di STMIK Antar Bangsa. Tahun 2020 lulus dari Program Pascasarjana (S2) Konsentrasi Teknologi Sistem Informasi pada Jurusan Ilmu Komputer di Universitas Budi Luhur. Saat ini aktif mengajar dan juga sebagai Operator PDDIKTI di Institut Daarul Qur'an Jakarta