

RANCANG BANGUN APLIKASI PENJADWALAN PERKULIAHAN PROGRAM STUDI TEKNIK INFORMATIKA DI UNIVERSITAS ISLAM BALITAR

Abimita Khasanatul Maghfiroh, Sri Lestanti, Saiful Nur Budiman

Program Studi Teknik Informatika, Universitas Islam Balitar

abimita10@gmail.com

ABSTRAK

Penjadwalan mata kuliah pada suatu perguruan tinggi merupakan hal yang sangat penting untuk diperhatikan, karena jadwal mata kuliah merupakan komponen yang menentukan berjalannya aktifitas perkuliahan dalam suatu perguruan tinggi. Permasalahan yang terjadi pada Program Studi Teknik Informatika Universitas Islam Balitar yaitu terbatasnya jumlah ruangan, jumlah mata kuliah yang beragam dan jumlah kelas yang banyak, sehingga membutuhkan kejelian ekstra dalam merancang penjadwalan perkuliahan. Oleh karena itu penelitian ini bertujuan untuk mempermudah Program Studi Teknik Informatika dalam merancang penjadwalan mata kuliah supaya tidak terjadi bentrok, untuk menyelesaikan permasalahan tersebut maka dibangun sebuah aplikasi penjadwalan mata kuliah menggunakan metode *Rapid Application Development* (RAD), dimana metode pengembangan perangkat lunak tersebut lebih menekankan pada kecepatan pengembangan, ketimbang kesempurnaan pada setiap detailnya. Aplikasi penjadwalan yang dibangun juga menerapkan algoritma genetika, algoritma genetika merupakan salah satu metode optimasi untuk permasalahan penjadwalan yang memiliki tahapan antara lain pembangkitan populasi awal, seleksi, crossover, dan mutasi. Berdasarkan hasil pengujian *black box*, *white box testing* dan validasi yang telah dilakukan, pengujian *black box testing* memperoleh persentase sebesar 97,7%, *white box testing* memperoleh nilai *Cyclomatic Complexity* sebesar 16 yang dikategorikan ke dalam *A more complex procedure* dan tingkat resiko *moderate*, dan validasi memperoleh hasil sebesar 77,5% sehingga sistem tersebut dikategorikan "Layak".

Kata kunci: Penjadwalan Mata Kuliah, Algoritma Genetika, Metode *Rapid Application Development*

1. PENDAHULUAN

Saat ini perkembangan teknologi semakin berkembang pesat, sehingga memberikan kemudahan bagi setiap orang. Kemajuan komputer juga berdampak pada kemajuan informasi, saat ini instansi pemerintahan maupun swasta sangat membutuhkan suatu teknologi komputerisasi sebagai alat yang berguna untuk mencari informasi dalam dunia pendidikan dan pekerjaan. Suatu aplikasi diperlukan oleh setiap orang untuk mengatasi permasalahan yang sederhana hingga permasalahan yang lebih kompleks [1].

Perencanaan dalam penjadwalan adalah bagian strategis dari proses mengendalikan urutan pekerjaan, mengalokasikan waktu, dan memfasilitasi operasi yang harus diselesaikan pada suatu organisasi, dengan proses waktu untuk menghasilkan output berdasarkan sumber daya yang ada dan dengan waktu yang diharapkan [2]. Penjadwalan kegiatan perkuliahan pada perguruan tinggi merupakan suatu hal yang sulit, dikarenakan dalam menyusun perkuliahan harus dipenuhi batasan - batasan tertentu, seperti ketersediaan ruangan, ketersediaan dosen, waktu perkuliahan, kebutuhan kurikulum, ketersediaan fasilitas, konflik jadwal antara mata kuliah dan jadwal kuliah dengan dosen tidak boleh sama dengan mata kuliah lain yang diajarkan oleh dosen lain dalam hal ruang dan waktu yang bersamaan. Keterbatasan tersebut membuat penyusunan jadwal perkuliahan menjadi sulit, dan jika banyak komponen yang harus dijadwalkan maka proses penjadwalan perkuliahan akan menjadi lebih rumit [3].

Program Studi Teknik Informatika di Universitas Islam Balitar memiliki tantangan tersendiri dalam penyusunan jadwal perkuliahan. Dengan jumlah kelas yang terbatas, jumlah mata kuliah yang beragam, dan ketersediaan dosen yang terbatas, perancangan jadwal perkuliahan menjadi suatu pekerjaan yang membutuhkan ketelitian dan keahlian khusus. Hal ini tidak hanya mempengaruhi efisiensi dalam pengelolaan perkuliahan, tetapi juga dapat berdampak pada pengalaman belajar mahasiswa.

Untuk mengatasi permasalahan ini, penulis melakukan penelitian dan membangun "Aplikasi Penjadwalan Perkuliahan Program Studi Teknik Informatika Di Universitas Islam Balitar." Aplikasi ini dirancang dengan memanfaatkan metode algoritma genetika yang memiliki keunggulan dalam mengoptimalkan permasalahan yang kompleks. Algoritma genetika bekerja melalui prinsip-prinsip teori evolusi, yang memungkinkan aplikasi ini menemukan solusi optimal dalam berbagai kasus, termasuk dalam penyusunan jadwal perkuliahan yang rumit [4].

Dalam proses pengembangan aplikasi ini, penulis menerapkan metode pengembangan *Rapid Application Development* (RAD) dan mengikuti Siklus Hidup Pengembangan Sistem (SDLC) untuk memastikan aplikasi berfungsi dengan baik. Metode tersebut memiliki kelebihan yaitu lebih menekankan siklus perancangan sistem yang singkat, cepat, dan tahapannya yang terstruktur, sehingga dalam pengerjaannya tidak membutuhkan waktu lama [1]. Selain itu, penelitian ini juga melibatkan pengujian

metode black box dan white box untuk memastikan bahwa aplikasi berjalan sesuai yang diharapkan tanpa kesalahan atau bug yang signifikan. Dengan dibangunnya aplikasi ini, diharapkan bahwa proses penjadwalan perkuliahan di Program Studi Teknik Informatika dapat menjadi lebih efisien dan tidak mengalami tumbukan yang mengganggu.

2. TINJAUAN PUSTAKA

2.1. Aplikasi

Aplikasi adalah sebuah program dalam komputer yang dirancang untuk menjalankan tugas tertentu dengan tujuan membantu pengguna dalam memenuhi kebutuhannya, dengan penggunaan aplikasi akan memperoleh hasil yang lebih akurat melalui pemrosesan data yang diharapkan. Aplikasi dapat dibuat untuk berbagai tujuan, seperti untuk keperluan bisnis, pendidikan, hiburan, atau produktivitas. Dalam pembuatan aplikasi melibatkan proses pengembangan yang kompleks, termasuk perancangan antarmuka pengguna, pengkodean, pengujian, dan peluncuran. Aplikasi yang baik adalah aplikasi yang dapat memenuhi kebutuhan penggunanya secara mudah, efektif, dapat berjalan dengan stabil dan aman digunakan di berbagai perangkat atau *platform* [5].

2.2. Penjadwalan

Penjadwalan adalah alokasi sumber daya berdasarkan waktu untuk menghasilkan serangkaian pekerjaan. Penjadwalan diperlukan untuk menentukan urutan kerja dengan alokasi sumber daya yang benar, proses penjadwalan menentukan bahan, mesin, dan operator yang dibutuhkan untuk menyelesaikan setiap pekerjaan. Dengan pengaturan penjadwalan yang diatur dengan baik perusahaan dapat memenuhi tenggat waktu dan standar untuk setiap pekerjaan [6].

Penjadwalan perkuliahan merupakan rutinitas yang penting dalam proses pembelajaran setiap semester. Dosen dan mahasiswa sangat berpengaruh dalam proses belajar mengajar, untuk menentukan struktur mata kuliah yang harus mengutamakan kepentingan dosen dan mahasiswa. Dalam menentukan penjadwalan ada beberapa batasan yang harus diperhatikan, jika batasan tersebut tidak dipertimbangkan dengan baik maka akan mempersulit proses penjadwalan [7].

2.3. Algoritma Genetika

Algoritma genetika adalah salah satu teknik dalam komputasi yang terinspirasi dari prinsip-prinsip genetika dan evolusi alami untuk memecahkan permasalahan optimasi, yaitu mencari solusi terbaik dari sekumpulan kemungkinan solusi yang tersedia. Algoritma genetika bekerja dengan cara memodelkan solusi masalah sebagai individu dalam sebuah populasi. Algoritma genetika telah digunakan dalam berbagai bidang, termasuk dalam masalah optimasi, desain mesin, kecerdasan buatan, dan pembelajaran mesin [8].

2.4. Hypertext Preprocessor (PHP)

PHP adalah *script* berbasis *server* yang terintegrasi dengan HTML (*server side HTML embedded scripting*). PHP dapat digunakan untuk membuat berbagai aplikasi berbasis web, mulai dari halaman web sederhana hingga program yang lebih kompleks dan membutuhkan koneksi database. PHP merupakan bahasa scripting yang paling banyak digunakan saat ini. PHP banyak digunakan untuk membuat *website* dinamis, dan dapat digunakan untuk tujuan yang lainnya [9].

2.5. MySQL

MySQL adalah perangkat lunak data berbasis *server* yang menggunakan sintaks *SQL* standar, memiliki basis pengguna yang besar, serta dapat memperoleh dan mengirimkan data dengan sangat cepat. MySQL didistribusikan di bawah lisensi GNU/GPL (*General Public License*). MySQL menggunakan bahasa *SQL* (*Structured Query Language*) yang menyediakan berbagai fitur dan kemampuan untuk mengakses, memanipulasi, dan menganalisis data, termasuk kemampuan untuk melakukan kueri (*query*) dan transaksi (*transaction*) data [2].

2.6. XAMPP

XAMPP adalah sebuah paket *software* untuk mengembangkan dan menjalankan aplikasi web di lingkungan lokal. XAMPP menyediakan beberapa aplikasi utama yang dibutuhkan untuk mengembangkan dan menjalankan aplikasi web, yaitu *web server* menggunakan Apache, sistem pengelola data menggunakan MySQL, bahasa pemrograman menggunakan PHP, dan Perl sebagai bahasa skrip. XAMPP bersifat *open source*, sehingga dapat digunakan secara gratis dan diubah sesuai dengan kebutuhan pengguna [7].

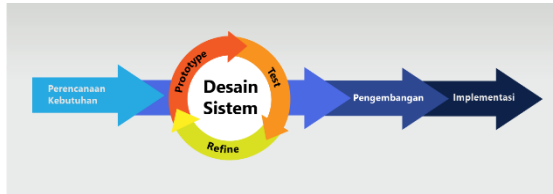
2.7. Metode Rapid Application Development (RAD)

Rapid Application Development (RAD) adalah sebuah metode pengembangan perangkat lunak yang berfokus pada pemakaian alat atau *platform* pengembangan perangkat lunak yang dapat mempercepat dan mempermudah proses pembuatan aplikasi sehingga lebih efektif dan efisien. RAD merupakan sebuah metode pengembangan perangkat lunak yang lebih menekankan pada kecepatan pengembangan, ketimbang kesempurnaan pada setiap detailnya, sehingga memungkinkan para pengembang untuk menghasilkan aplikasi dalam waktu yang relatif singkat [10].

3. METODE PENELITIAN

Metode pengembangan yang digunakan dalam penelitian ini adalah metode pengembangan model RAD (*Rapid Application Development*). Metode RAD (*Rapid Application Development*) dapat diterapkan pada jenis penelitian pengembangan (R&D) dengan tujuan untuk menghasilkan produk perangkat lunak

yang lebih cepat, lebih efektif, dan lebih efisien. Metode RAD dapat membantu peneliti untuk mempercepat proses pengembangan produk perangkat lunak dengan menggunakan pendekatan iteratif dan incremental [10]. Tahapan pada metode ini dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Metode RAD

3.1. Perencanaan Kebutuhan

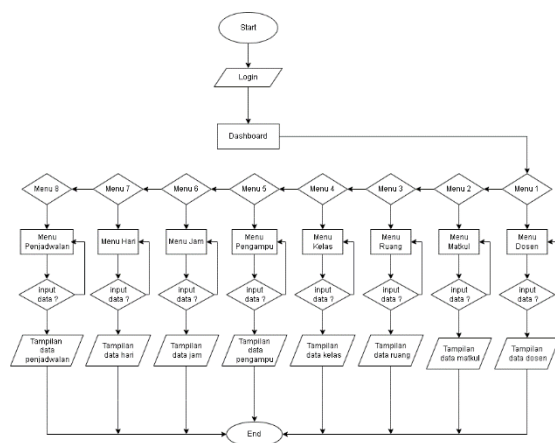
Pada tahap pertama, hal yang perlu dilakukan adalah analisa kebutuhan sistem untuk aplikasi penjadwalan. Adapun kebutuhan sistem tersebut seperti komputer/laptop dengan processor minimal Core i3, RAM minimal 4GB, dan hardisk minimal 250GB.

3.2. Desain Sistem

Pada tahap kedua memiliki 3 poin penting dalam proses pengembangan sistem, diantaranya:

a. Prototype

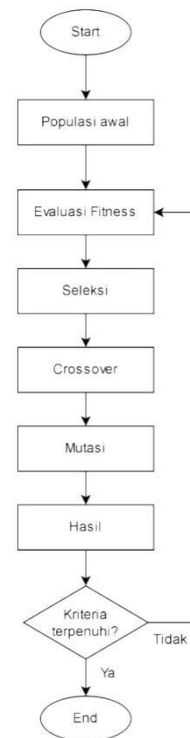
Prototype dibuat sebagai langkah awal dalam pengembangan produk, dan dapat membantu dalam pengujian, perbaikan, serta evaluasi ide dan konsep. Sebuah prototype dapat dibuat untuk menguji fitur dan fungsionalitas, mendapatkan umpan balik dari pengguna atau stakeholder, serta untuk mempercepat proses pengembangan dengan meminimalisir kesalahan yang mungkin terjadi di tahap akhir.



Gambar 2. Flowchart Sistem Penjadwalan

Pada Gambar 2 merupakan *flowchart* sistem penjadwalan yang menggambarkan proses berjalannya aplikasi setelah admin berhasil login. Terdapat 8 menu pada halaman dashboard, yaitu menu data dosen, data mata kuliah, data ruang, data kelas, data pengampu, data jam, data hari, dan data penjadwalan. Semua data

pada menu tersebut dapat diakses dan dikelola oleh kaprodi.



Gambar 3. Flowchart Generate Sistem Penjadwalan

Pada Gambar 3. merupakan *flowchart generate* sistem penjadwalan. Proses *generate* penjadwalan menggunakan algoritma genetika dimulai dari populasi awal, dimana sekumpulan beberapa individu digunakan sebagai basis dalam evolusi algoritma genetika. Selanjutnya fungsi *fitness* dihitung berdasarkan jumlah pelanggaran atau penalty yang terjadi pada setiap slot kromosom dalam suatu individu. Seleksi individu dilakukan berdasarkan nilai *fitness* masing – masing untuk memilih individu mana yang akan mengalami proses perkawinan atau pindah silang. Metode crossover, dimana tidak semua induk akan mengalami proses crossover, karena pemilihan induk dipilih secara acak sebanyak crossover rate yang sudah diatur di awal. Kemudian proses mutasi menggunakan skema *swap mutation*. Jumlah kromosom yang mengalami mutasi dalam satu populasi ditentukan oleh *Mutation Rate (MR)*.

b. Test

Tahapan ini merupakan penyesuaian desain prototype dengan keperluan pengguna, maka akan dilakukan perbaikan prototype sesuai dengan keinginan pengguna.

c. Refine

Tahapan refine merupakan tahapan penyempurnaan setelah dilakukan pengujian pada desain prototype. Tujuan dari tahap refine adalah untuk memperbaiki dan meningkatkan kualitas agar

dapat lebih baik lagi dalam memenuhi kebutuhan pengguna.

3.3. Pengembangan

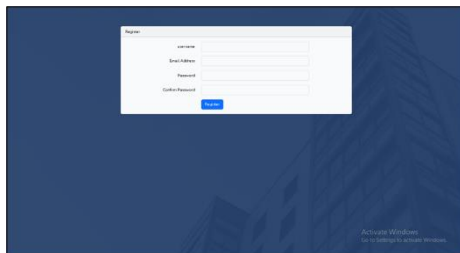
Pada tahap ketiga, dilakukan pengembangan aplikasi perangkat lunak menggunakan bahasa pemrograman PHP, dengan *framework Laravel* dan *framework bootstrap*. Tools yang digunakan adalah *visual studio code* dan *xampp* yang digunakan sebagai alat untuk pengembangan aplikasi penjadwalan perkuliahan.

3.4. Implementasi

Pada tahap ini akan dilakukan pengujian aplikasi berbasis skenario dengan menggunakan metodologi pengujian *black box*, *white box testing*, dan validasi. Prosedur ini untuk memastikan bahwa fungsionalitas sistem telah di implementasikan dengan hati-hati untuk memudahkan pengembang mengenali masalah sistem saat prosedur sedang dilakukan.

4. HASIL DAN PEMBAHASAN

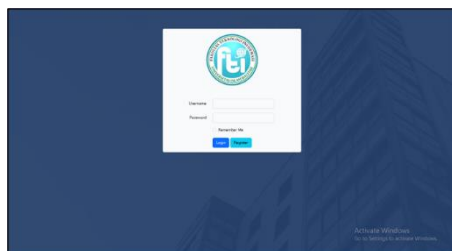
4.1. Tampilan Halaman Register



Gambar 4. Tampilan Halaman Register

Pada gambar 4 menggambarkan tampilan halaman *register*. Halaman *register* digunakan *user* untuk mendaftarkan akun agar bisa mengakses *website* tersebut.

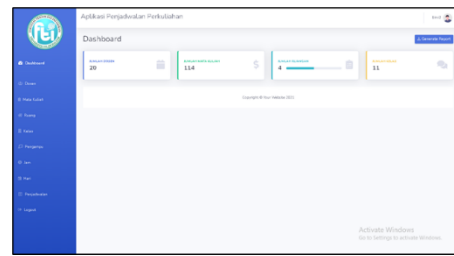
4.2. Tampilan Halaman Login



Gambar 5. Tampilan Halaman Login

Pada gambar 5 menggambarkan tampilan halaman *login*. Halaman *login* digunakan *user* untuk masuk ke halaman *dashboard*, sebelum *login user* (kaprodi) harus mengisi *username* dan *password* yang sudah terdaftar.

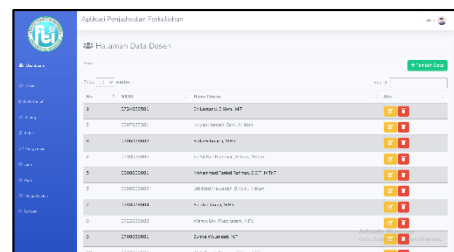
4.3. Tampilan Halaman Dashboard



Gambar 6. Tampilan Halaman Dashboard

Pada gambar 6 menggambarkan tampilan halaman *dashboard admin*. Halaman *dashboard* tersebut tidak memiliki interaksi apapun dengan *user*, hanya berupa tampilan informasi yang berupa jumlah dosen, jumlah mata kuliah, jumlah ruangan dan jumlah kelas yang terdapat pada aplikasi tersebut.

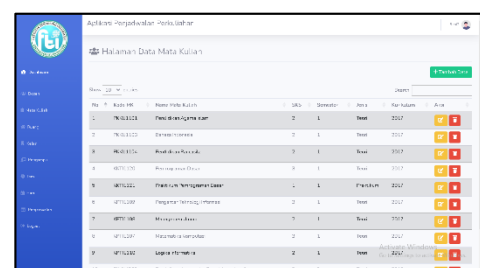
4.4. Tampilan Menu Dosen



Gambar 7. Tampilan Menu Dosen

Pada gambar 7 menggambarkan tampilan menu dosen. Halaman menu dosen berupa tabel yang berisi sebuah informasi mengenai data dosen yang telah di inputkan pada form tabel tersebut. Halaman data dosen berisi NIDN dan nama dosen yang mengajar di Program Studi Teknik Informatika.

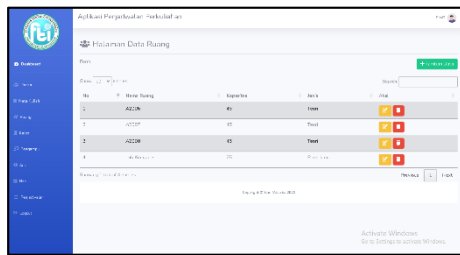
4.5. Tampilan Menu Mata Kuliah



Gambar 8. Tampilan Menu Mata Kuliah

Pada gambar 8 menggambarkan tampilan menu mata kuliah. Halaman menu mata kuliah berisi sebuah informasi mengenai data mata kuliah yang telah di inputkan pada form tabel tersebut.

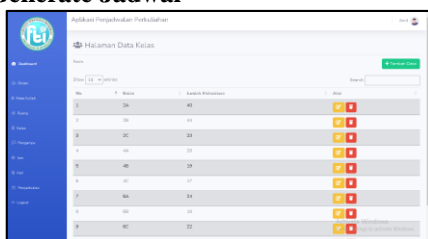
4.6. Tampilan Menu Ruang



Gambar 9. Tampilan Halaman Generate Jadwal

Pada gambar 9 menggambarkan tampilan menu ruang. Halaman menu ruang berisi sebuah informasi mengenai data ruang yang telah di inputkan pada form tabel tersebut.

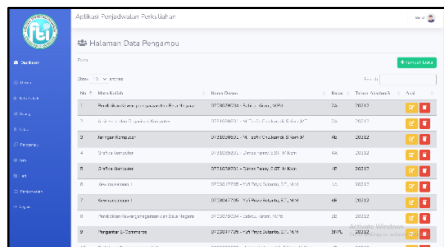
4.7. Generate Jadwal



Gambar 10. Tampilan Menu Kelas

Pada gambar 10 menggambarkan tampilan menu kelas. Halaman menu kelas berisi sebuah informasi mengenai data kelas yang telah di inputkan pada form tabel tersebut.

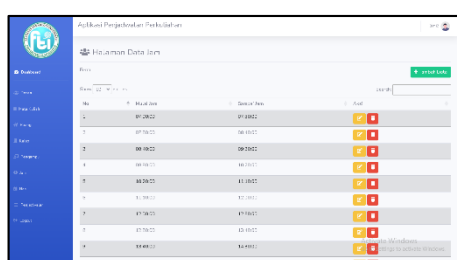
4.8. Tampilan Menu Pengampu



Gambar 11. Tampilan Menu Pengampu

Pada gambar 11 menggambarkan tampilan menu pengampu. Halaman menu pengampu berisi sebuah informasi mengenai data pengampu yang telah di inputkan pada form tabel tersebut.

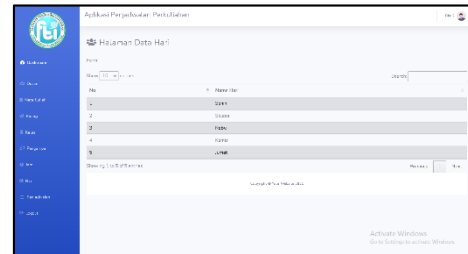
4.9. Tampilan Menu Waktu Perkuliahan



Gambar 12. Tampilan Menu Waktu Perkuliahan

Pada gambar 12 menggambarkan tampilan menu waktu perkuliahan. Halaman menu waktu perkuliahan berisi range waktu perkuliahan pertama sampai waktu perkuliahan terakhir.

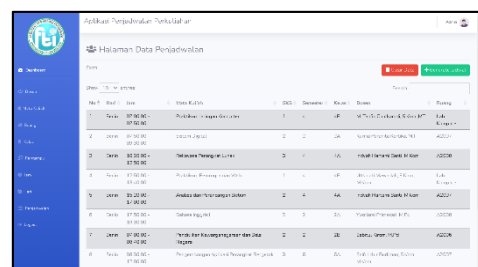
4.10. Tampilan Menu Hari



Gambar 13. Tampilan Menu Hari

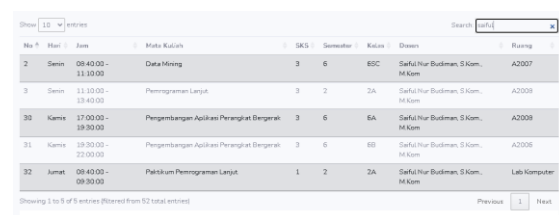
Pada gambar 13 menggambarkan tampilan menu hari. Halaman menu hari berisi nama – nama hari yang digunakan perkuliahan dalam satu minggu. Menu hari tersebut digunakan sebagai acuan dalam melakukan generate jadwal.

4.11. Tampilan Menu Penjadwalan



Gambar 14. Tampilan Menu Penjadwalan

Pada gambar 14 menggambarkan tampilan menu penjadwalan. Halaman menu penjadwalan merupakan halaman yang digunakan untuk menentukan penjadwaalan atau mengenerate jadwal mata kuliah menggunakan algoritma genetika. Hasil generate dari aplikasi penjadwalan ini memiliki 3 ketentuan yang harus terpenuhi, yang pertama yaitu setiap dosen dalam satu hari minimal mengajar sebanyak 2x dan maksimal mengajar sebanyak 4x. Kedua setiap kelas dalam sehari maksimal 3x mata kuliah. Ketiga dihari jumat waktu perkuliahan dimulai pukul 08.40, untuk lebih jelasnya akan di paparkan pada gambar dibawah ini.



Gambar 15. Hasil Generate Ketentuan Dosen

Gambar 15 merupakan hasil generate penjadwalan yang membuktikan bahwa setiap dosen dalam satu hari minimal mengajar sebanyak 2x dan maksimal mengajar 4x. Sebagai contoh pada gambar 4.27 hasil generate ketentuan dosen atas nama bapak saiful nur budiman pada hari senin mengajar sebanyak 2x, hari kamis mengajar sebanyak 2x dan dihari jumat mengajar sebanyak 1x.

4.12. Pengujian Blackbox Testing

Setelah proses pemrograman selesai dibuat, maka tahap selanjutnya adalah pengujian sistem menggunakan *black box testing* yaitu melakukan *test-case* terhadap aplikasi dengan menggunakan tabel pengujian yaitu dengan cara memasukkan data ke dalam sistem dan melihat hasil keluarannya apakah sesuai dengan hasil yang diharapkan atau tidak.

Table 1. Test Case Form Register

No	Skenario Pengujian	Test Case	Hasil yang Diinginkan	Hasil dari Pengujian	Kesimpulan
1.	Mengisi form register dengan benar	Username: Admin Email : admin@gmail.com Password: admin123 konfirmasi password : admin123	Jika registrasi berhasil akan diarahkan ke halaman dashboard	Register telah berhasil dan diarahkan ke halaman dashboard	Sesuai
2.	Mengisi password register kurang dari 8 karakter	Username: Admin Email : admin@gmail.com Password: admin1 konfirmasi password : admin1	Akan muncul notifikasi bahwa password harus minimal 8 karakter	Sistem memberikan notifikasi password harus minimal 8 karakter	Sesuai
3.	Mengisi username atau email yang sudah terdaftar di database	Username: Admin (sudah terdaftar) Email : admin@gmail.com (sudah terdaftar)	Muncul notifikasi bahwa username atau email tersebut telah terdaftar	Sistem memberikan notifikasi bahwa username atau email tersebut telah terdaftar	Sesuai
4.	Mengisi password dan konfirmasi password tidak sama	Username: Admin Email : admin@gmail.com Password: admin123 Konfirmasi password : admin12	Muncul notifikasi bahwa password tidak sama	Sistem memberikan notifikasi bahwa password dan konfirmasi password harus sama	Sesuai

Pengujian test case form register dilakukan sebanyak 4 kali. Hasil pengujian 4 fungsi tersebut dihitung persentasenya sesuai rumus sebagai berikut.

$$x = \frac{4}{4} \times 100\% = 100\%$$

Tabel 2. Test Case Form Login

No	Skenario Pengujian	Test Case	Hasil yang Diinginkan	Hasil dari Pengujian	Kesimpulan
1.	Masukkan username dan password yang benar	Username : admin (benar) Password : admin123 (benar)	Login akan berhasil dan masuk halaman utama	Login berhasil dan masuk halaman utama	Sesuai
2.	Masukkan username dan password yang salah	Username : Admin1 (salah) Password : admin1 (salah)	Muncul notifikasi bahwa login gagal, username dan password salah	Muncul notifikasi bahwa login gagal, username dan password salah	Sesuai

Pengujian *test case* form login dilakukan sebanyak 2 kali. Pertama pengujian pada fungsi login admin, jika sudah memasukkan email dan password yang benar, login akan berhasil. Kedua pengujian pada fungsi kegagalan login, jika sudah memasukkan email dan password yang salah, maka login akan gagal. Hasil

dari pengujian 2 fungsi tersebut dihitung persentasenya sesuai rumus sebagai berikut :

$$x = \frac{2}{2} \times 100\% = 100\%$$

Tabel 3. Test Case Form Dosen

No	Skenario Pengujian	Test Case	Hasil yang Diinginkan	Hasil dari Pengujian	Kesimpulan
1.	Admin menambahkan data dosen	NIDN: 0987654321 Nama dosen : Alfarizi	Data akan ditambahkan dan tersimpan di <i>database</i>	Data berhasil di tambahkan dan terlihat di halaman data dosen	Sesuai
2.	Admin mengedit data dosen	NIDN: 0987654321 Nama dosen : Alfarizi → (M. Alfarizi)	Nama dosen berhasil diubah dan tersimpan	Data otomatis berubah	Sesuai

Pengujian *test case* form dosen dilakukan sebanyak 3 kali. Pertama pengujian pada fungsi tambah data, Kedua pengujian pada fungsi edit data, Ketiga pengujian pada fungsi hapus. Hasil dari pengujian 3 fungsi tersebut dihitung persentasenya sesuai rumus sebagai berikut :

$$x = \frac{3}{3} \times 100\% = 100\%$$

Tabel 4. Test Case Form Mata Kuliah

No	Skenario Pengujian	Test Case	Hasil yang Diinginkan	Hasil dari Pengujian	Kesimpulan
1.	Admin menambahkan data mata kuliah	Kode mata kuliah : KKKK1234 Nama mata kuliah : PKN SKS : 2 Semester : 2 Jenis : Teori Kurikulum : 2017	Data akan ditambahkan dan tersimpan di <i>database</i>	Data berhasil di tambahkan dan terlihat di halaman data mata kuliah	Sesuai
2.	Admin mengedit data mata kuliah	Kode mata kuliah : KKKK1234 → ABCD1234 Nama mata kuliah : PKN → Matematika SKS : 2 Semester : 2 Jenis : Teori Kurikulum : 2017	Data berhasil diubah dan tersimpan	Data otomatis berubah	Sesuai

Pengujian *test case* form mata kuliah dilakukan sebanyak 3 kali. Pertama pengujian pada fungsi tambah data, kedua pengujian fungsi edit data dan ketiga fungsi hapus data. Hasil dari pengujian 3 fungsi tersebut dihitung persentasenya sesuai rumus sebagai berikut :

$$x = \frac{3}{3} \times 100\% = 100\%$$

Tabel 5. Test Case Form Ruang

No	Skenario Pengujian	Test Case	Hasil yang Diinginkan	Hasil dari Pengujian	Kesimpulan
1.	Admin menambahkan data ruang	Nama ruang : A2011 Kapasitas : 35 Jenis : Teori	Data akan ditambahkan dan tersimpan di <i>database</i>	Data berhasil di tambahkan dan terlihat di halaman data ruang	Sesuai
2.	Admin mengedit data ruang	Nama ruang : A2011 → B2011 Kapasitas : 35 → 32 Jenis : Teori	Data berhasil diubah dan tersimpan	Data otomatis berubah	Sesuai
3.	Admin menghapus data ruang	Hapus data	Sebelum menghapus data, ada notifikasi “yakin hapus data?”	Sistem memberikan notifikasi “yakin hapus data?”	Sesuai

Pengujian *test case* form ruang dilakukan sebanyak 3 kali. Hasil dari pengujian 3 fungsi tersebut dihitung persentasenya sesuai rumus sebagai berikut :

$$x = \frac{3}{3} \times 100\% = 100\%$$

Tabel 6. Test Case Form Kelas

No	Skenario Pengujian	Test Case	Hasil yang Diinginkan	Hasil dari Pengujian	Kesimpulan
1.	Admin menambahkan data kelas	Kelas : 2D Jumlah mahasiswa : 27	Data akan ditambahkan dan tersimpan di <i>database</i>	Data berhasil di tambahkan dan terlihat di halaman data kelas	Sesuai
2.	Admin mengedit data kelas	Kelas : 2D → 4D	Data berhasil diubah dan tersimpan	Data otomatis berubah	Sesuai

No	Skenario Pengujian	Test Case	Hasil yang Diinginkan	Hasil dari Pengujian	Kesimpulan
		Jumlah mahasiswa : 27 → 30			
3.	Admin menghapus data kelas	Hapus data	Sebelum menghapus data, ada notifikasi “yakin hapus data?”	Sistem memberikan notifikasi “yakin hapus data?”	Sesuai

Pengujian *test case* form kelas dilakukan sebanyak 3 kali. Hasil dari pengujian 3 fungsi tersebut dihitung persentasenya sesuai rumus sebagai berikut :

$$x = \frac{3}{3} \times 100\% = 100\%$$

Tabel 7. Test Case Form Jam

No	Skenario Pengujian	Test Case	Hasil yang Diinginkan	Hasil dari Pengujian	Kesimpulan
1.	Admin menambahkan data jam	Mula jam : 08.00 Sampai jam : 08.50	Data akan ditambahkan dan tersimpan di <i>database</i>	Data berhasil di tambahkan dan terlihat di halaman data jam	Sesuai
2.	Admin mengedit data jam	Mula jam : 08.00 → 08.10 Sampai jam : 08.50 → 09.00	Data berhasil diubah dan tersimpan	Data otomatis berubah	Sesuai
3.	Admin menghapus data jam	Hapus data	Sebelum menghapus data, ada notifikasi “yakin hapus data?”	Sistem memberikan notifikasi “yakin hapus data?”	Sesuai

Pengujian *test case* form jam dilakukan sebanyak 3 kali. Hasil dari pengujian 3 fungsi tersebut dihitung persentasenya sesuai rumus sebagai berikut :

$$x = \frac{3}{3} \times 100\% = 100\%$$

Tabel 8. Test Case Form Pengampu

No	Skenario Pengujian	Test Case	Hasil yang Diinginkan	Hasil dari Pengujian	Kesimpulan
1.	Admin menambahkan data pengampu	Mata kuliah : Pilih opsi matkul Nama dosen : Pilih opsi dosen Kelas : Pilih opsi kelas Tahun akademik : 20212	Data akan ditambahkan dan tersimpan di <i>database</i>	Data berhasil di tambahkan dan terlihat di halaman data pengampu	Sesuai
2.	Admin mengedit data pengampu	Mata kuliah : Pilih opsi matkul Nama dosen : Pilih opsi dosen Kelas : Pilih opsi kelas Tahun akademik : 20211	Data berhasil diubah dan tersimpan	Data otomatis berubah	Sesuai
3.	Admin menghapus data pengampu	Hapus data	Sebelum menghapus data, ada notifikasi “yakin hapus data?”	Sistem memberikan notifikasi “yakin hapus data?”	Sesuai

Pengujian *test case* form pengampu dilakukan sebanyak 3 kali. Hasil dari pengujian 3 fungsi tersebut dihitung persentasenya sesuai rumus sebagai berikut :

$$x = \frac{3}{3} \times 100\% = 100\%$$

Tabel 9. Test Case Form Penjadwalan

No	Skenario Pengujian	Test Case	Hasil yang Diinginkan	Hasil dari Pengujian	Kesimpulan
1.	Generate penjadwalan	Klik <i>button</i> generate	Ketika tombol button klik maka sistem akan mengenetare jadwal	Sistem mengenerate date dan menghasilkan output penjdwalan perkuliahan	Sesuai
2.	Search data	Search salah satu nama dosen	Akan terlihat data dosen dimana dalam sehari	Menampilkan hasil data dosen dalam sehari minimal mengajar	Tidak Sesuai

No	Skenario Pengujian	Test Case	Hasil yang Diinginkan	Hasil dari Pengujian	Kesimpulan
			minimal mengajar 2x dan maksimal mengajar 4x.	sebanyak 1x dan maksimal mengajar sebanyak 4x	
3.	Search data	Search salah satu nama kelas	Akan terlihat data kelas dimana setiap kelas dalam sehari maksimal 3 mata kuliah.	Menampilkan hasil data kelas dimana setiap kelas dalam sehari maksimal 3 mata kuliah.	Sesuai
4.	Search data	Search hari jumat	Akan terlihat data dimana hari jumat perkuliahan dimulai pukul 08.40	Menampilkan data hari jumat perkuliahan dimulai pukul 08.40	Sesuai
5.	Menghapus data penjadwalan	Clear data	Data akan terhapus	Sistem berhasil menghapus data	Sesuai

Pengujian *test case* form penjadwalan dilakukan sebanyak 5 kali. Hasil dari pengujian 5 fungsi tersebut dihitung persentasenya sesuai rumus sebagai berikut :

$$x = \frac{4}{5} \times 100\% \\ = 80\%$$

Jadi hasil persentase *black box testing* secara keseluruhan adalah sebesar 97,7%.

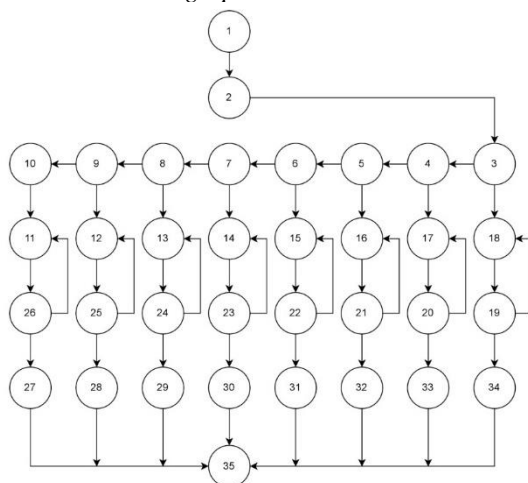
4.13. Whitebox Testing

Pengujian white box testing berfungsi untuk menguji sebuah software tanpa harus memperhatikan detail dari sebuah software tersebut. Berikut dibawah ini alur dari pengujian white box testing pada Aplikasi Penjadwalan Perkuliahan Program Studi Teknik Informatika di Universitas Islam Balitar menggunakan basis path.

a. Analisis Flowchart

Hal pertama dalam pengujian whitebox testing yaitu melakukan analisis terhadap flowchart sistem yang telah dirancang. Flowchart tersebut tersimpan pada bab 3 gambar 3.5. Selanjutnya flowchart tersebut akan diubah menjadi flowgraph seperti berikut:

b. Membuat Flowgraph



Gambar 16 Notasi Alir (flowgraph)

c. Penentuan Independent Path berdasarkan perhitungan Cyclomatic Complexity

Cyclomatic Complexity merupakan *metric* yang mendefinisikan jumlah alur independen dalam *basis* set suatu program dan menyediakan batas atas untuk jumlah pengujian yang harus dilakukan guna memastikan bahwa semua pernyataan telah di eksekusi minimal sekali. Untuk menghitung *Cyclomatic Complexity* digunakan rumus sebagai berikut :

$$V(G) = E - N + 2$$

Dari flowgraph diatas, sehingga memperoleh hasil *Cyclomatic Complexity* sebagai berikut :

$$V(G) = E - N + 2$$

$$V(G) = 49 - 35 + 2$$

$$V(G) = 16$$

Berdasarkan nilai perhitungan *Cyclomatic Complexity* diatas, maka jalur independent yang didapat antara lain ;

- 1) 1-2-3-18-19-34-35
- 2) 1-2-3-18-19-18-19-34-35
- 3) 1-2-3-4-17-20-23-35
- 4) 1-2-3-4-17-20-17-20-33-35
- 5) 1-2-3-4-5-16-21-32-35
- 6) 1-2-3-4-5-16-21-16-21-32-35
- 7) 1-2-3-4-5-6-15-22-31-35
- 8) 1-2-3-4-5-6-15-22-15-22-31-35
- 9) 1-2-3-4-5-6-7-14-23-30-35
- 10) 1-2-3-4-5-6-7-14-23-14-23-30-35
- 11) 1-2-3-4-5-6-7-8-13-24-29-35
- 12) 1-2-3-4-5-6-7-8-13-24-13-24-29-35
- 13) 1-2-3-4-5-6-7-8-9-12-25-28-35
- 14) 1-2-3-4-5-6-7-8-9-12-25-12-25-28-35
- 15) 1-2-3-4-5-6-7-8-9-10-11-26-27-35
- 16) 1-2-3-4-5-6-7-8-9-10-11-26-11-26-27-35

Hasil perhitungan *cyclomatic complexity* yang diperoleh sebesar 16.

4.14. Validasi

Pengujian validasi dilakukan kepada ahli dan pengguna, ahli IT oleh bapak Mohammad Faried Rahmat selaku dosen IT dan validasi pengguna oleh kaprodi program studi teknik informatika Universitas

Islam Balitar menggunakan kuesioner, sehingga memperoleh hasil sebagai berikut:

Tabel 10 Hasil Kuesioner

No.	Pertanyaan	Penilaian				
		SS	S	C	K	T
1.	Bagaimana pendapat anda mengenai tampilan antar muka aplikasi penjadwalan perkuliahan ini?	-	2	-	-	-
2.	Bagaimana pendapat anda mengenai penggunaan bahasa dan kata pada aplikasi ini ?	-	1	1	-	-
3.	Bagaimana pendapat anda mengenai kejelasan tulisan pada tabel aplikasi ini ?	-	2	-	-	-
4.	Bagaimana pendapat anda mengenai tampilan menu pada aplikasi ini ?	1	1	-	-	-
5.	Bagaimana pendapat anda mengenai fitur pada aplikasi ini ?	-	1	1	-	-
6.	Bagaimana pendapat anda mengenai proses CRUD pada aplikasi ini ?	-	2	-	-	-
7.	Apakah aplikasi penjadwalan ini mudah dipahami ?	-	2	-	-	-
8.	Apakah aplikasi ini membantu dalam proses penjadwalan perkuliahan ?	-	1	1	-	-
JUMLAH		1	12	3	0	0

Tabel 10 merupakan jumlah kategori setiap pertanyaan yang akan dihitung jumlah total dari keseluruhan nilai kategori yang telah diisi. Jumlah dari tabel diatas digunakan untuk menghitung kelayakan dari aplikasi yang telah dibuat.

Tabel 11 Hasil Tabulasi Kuesioner

Nilai Penilaian	Jumlah Penilaian	Total (Skala×Jumlah)
5	1	5
4	12	48
3	3	9
2	0	0
1	0	0
Total Skor		62
Skor Maksimum		80

$$\text{Persentase skor total} = \frac{62}{80} \times 100\% = 77,5\%$$

Berdasarkan hasil pengolahan data yang telah dilakukan dari hasil kuesioner, maka memperoleh hasil perhitungan secara keseluruhan yaitu 77,5%.

Tabel 12 Kategori Kelayakan Sistem

No.	Kategori	Skor dalam Persentase
1	Sangat Layak	81% - 100%
2	Layak	61% - 80%
3	Cukup Layak	41% - 60%
4	Tidak Layak	21% - 40%
5	Sangat Tidak Layak	-21%

Setelah memperoleh hasil dari semua pengujian, tahapan selanjutnya adalah menghitung kelayakan sistem dengan cara berikut:

$$\text{Persentase (\%)} = \frac{\text{Nilai Yang Didapat}}{\text{Nilai Maksimal}} \times 100\%$$

$$\text{Persentase (\%)} = \frac{97,7 + 16 + 77,5}{300} \times 100\%$$

$$\text{Persentase (\%)} = \frac{191,2}{300} \times 100\%$$

$$\text{Persentase (\%)} = 63,73\%$$

Hasil persentase 63,73% tersebut dikategorikan sebagai sistem yang “Layak” (Sumber: Mandala & Dewanto, 2017)

5. KESIMPULAN DAN SARAN

Perancangan aplikasi penjadwalan perkuliahan di program studi teknik informatika menggunakan framework laravel dan database php MyAdmin yang di dalam aplikasinya menampilkan halaman register, halaman login, halaman dashboard, halaman dosen, halaman mata kuliah, halaman ruang, halaman kelas, halaman pengampu, halaman jam, halaman hari, dan halaman penjadwalan. Aplikasi penjadwalan perkuliahan ini juga menerapkan algoritma genetika yang terletak pada halaman menu penjadwalan. Hasil pengujian dari aplikasi penjadwalan perkuliahan di program studi teknik informatika mendapatkan hasil yang dapat dikategorikan “Layak”. Dimana kelayakan tersebut diperoleh berdasarkan pengujian black box, white box testing dan validasi. Pengujian black box testing memperoleh hasil sebesar 97,7%, white box testing memperoleh hasil perhitungan sebesar 16 yang dikategorikan sebagai A more complex procedure dengan tingkat resiko moderate dan hasil perhitungan validasi sebesar 77,5. Berdasarkan hasil dari pembahasan, terdapat beberapa saran yang di harapkan dari peneliti yaitu penambahan fitur seperti fitur cetak penjadwalan, perbaikan sistem pada ketentuan minimal dan maksimal setiap dosen mengajar dalam satu hari, dan dilakukannya pembaharuan pada sebagian program sistem atau melengkapi kelemahan dari program secara bertahap.

DAFTAR PUSTAKA

- [1] Sagala, J. R. (2018). Model Rapid Application Development (Rad) Dalam Pengembangan Sistem Informasi Penjadwalanbelajar Mengajar. *Jurnal Mantik Penusa*, 2(1), 88.[1] Jijon Raphita Sagala, “Model Rapid Application Development (Rad) Dalam Pengembangan Sistem Informasi Penjadwalanbelajar Mengajar,”

- J. Mantik Penusa*, vol. 2, no. 1, p. 88, 2021.
- [2] I. Anugrah, "Pengembangan Sistem Informasi Penjadwalan Mata Kuliah Berbasis Web Di Fakultas Teknik Universitas Negeri Makassar," *J. Pendidik.*, vol. 6, no. 2, 2018.
- [3] F. Ayu and W. Sholeha, "Rancang Bangun Sistem Informasi Penjadwalan Mata Pelajaran Berbasis Web Pada Smart Center Pekanbaru," *J. Intra-Tech*, vol. 3, no. 1, pp. 48–59, 2019.
- [4] Y. Elva, "Sistem Penjadwalan Mata Pelajaran Menggunakan Algoritma Genetika," *J. Teknol. Inf.*, vol. 3, no. 1, p. 49, 2019, doi: 10.36294/jurti.v3i1.687.
- [5] I. M. Widiarta, M. Julkarnain, and J. Imanulloh, "Rancang Bangun Aplikasi Uts in Me Berbasis Android Menggunakan Flutter Dengan Metode Rapid Application Development," *J. Inform. Teknol. dan Sains*, vol. 3, no. 4, pp. 447–452, 2021, doi: 10.51401/jinteks.v3i4.1323.
- [6] I. M. B. Adnyana and I. K. Wijayana, "Rancang Bangun Sistem Penjadwalan Sidang Skripsi Menggunakan Algoritma Genetika," *J. Sist. dan Inform.*, vol. 12, no. 1, pp. 38–47, 2017.
- [7] I. Ivy and D. O. Oktarina, "Sistem Informasi Penjadwalan Perkuliahan," *J. Mhs. Apl. Teknol. Komput. ...*, vol. 3, no. 1, pp. 1–6, 2021, [Online]. Available: <http://www.ejournal.pelitaindonesia.ac.id/JMapTeKsi/index.php/JOM/article/view/594>
- [8] A. Assagaf, A. Ibrahim, and C. Suranto, "Membangun Sistem Informasi Penjadwalan Dengan Metode Algoritma Genetika Pada Laboratorium Teknik Informatika Universitas Muhammadiyah Maluku Utara," *J. Ilm. Ilk. - Ilmu Komput. Inform.*, vol. 1, no. 2, pp. 95–105, 2018, doi: 10.47324/ilkominfo.v1i2.13.
- [9] A. S. Kusuma and I. G. S. E. Putra, "Rancang Bangun Sistem Penjadwalan Wawancara Mahasiswa Baru STMIK STIKOM Indonesia," *J. Ilm. Teknol. Inf. Asia*, vol. 11, no. 2, p. 139, 2017, doi: 10.32815/jitika.v11i2.202.
- [10] R. Kaban and R. J. Nasution, "Penerapan Metode Rapid Application Development (RAD) dalam Perancangan Sistem Pemesanan Menu menggunakan Quick Response (QR) Code," *MEANS (Media Inf. Anal. dan Sist.*, vol. 5, no. 2, pp. 144–152, 2020, doi: 10.54367/means.v5i2.920.