

BAB 3 METODOLOGI PENELITIAN

3.1 Metode Penelitian

Metode Penelitian yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Identifikasi Masalah

Identifikasi masalah merupakan tahap pertama dari penelitian yang dilakukan dengan mengidentifikasi dan merumuskan masalah untuk dijadikan sebagai pedoman penelitian secara keseluruhan.

2. Studi Literatur

Telaah literatur merupakan tahap penelitian yang bertujuan untuk mencari informasi lebih lanjut terkait dengan topik yang telah ditentukan. Pengumpulan informasi tersebut dilakukan dengan memahami bacaan dari berbagai sumber yang ditemukan seperti jurnal ilmiah, buku, *conference paper*, *research paper*, dan sumber-sumber lainnya.

3. Perancangan Sistem

Perancangan sistem dimulai dengan pembuatan rancangan *flowchart*, perancangan *interface website*, serta perancangan struktur data dari penelitian.

4. Pembuatan Program

Pembuatan program atau *website* berdasarkan *flowchart* dan *interface* yang telah dibuat dan diteruskan dengan mengimplementasikan algoritma pada program.

5. Testing dan Evaluasi

Tahap *testing* dilakukan untuk menguji tingkat akurasi program yang telah dibangun apabila dibandingkan dengan hasil pengecekan manual. Dalam penelitian ini, metode *testing* yang akan digunakan adalah metode *black box testing*. Setelah *testing* berhasil mendapatkan hasil yang memuaskan, maka hasil *testing* tersebut dievaluasi dengan memberikan tautan untuk mengakses *website* serta *quisioner* terkait hasil melalui media sosial. Adapun

metode yang digunakan untuk mengukur tingkat kepuasan mahasiswa dalam menggunakan sistem pada *website* yang telah dibangun adalah metode pengukuran EUCS (*End User Computing Satisfaction*). Dari akses *website* yang telah dibagikan serta *quisioner* dalam bentuk *Google Form*, responden diminta untuk menilai serta memberi tanggapan akan kepuasan terhadap penggunaan sistem penjadwalan tersebut. Pengambilan sampel yang digunakan pada penelitian ini merujuk pada metode penelitian menurut Sugiyono.

6. Penulisan Laporan

Penulisan laporan dilakukan sebagai bentuk dokumentasi dari penelitian, perancangan, serta pembuatan sistem. Laporan ini diharapkan untuk dapat menjadi bahan referensi untuk penelitian yang serupa di masa yang akan datang.

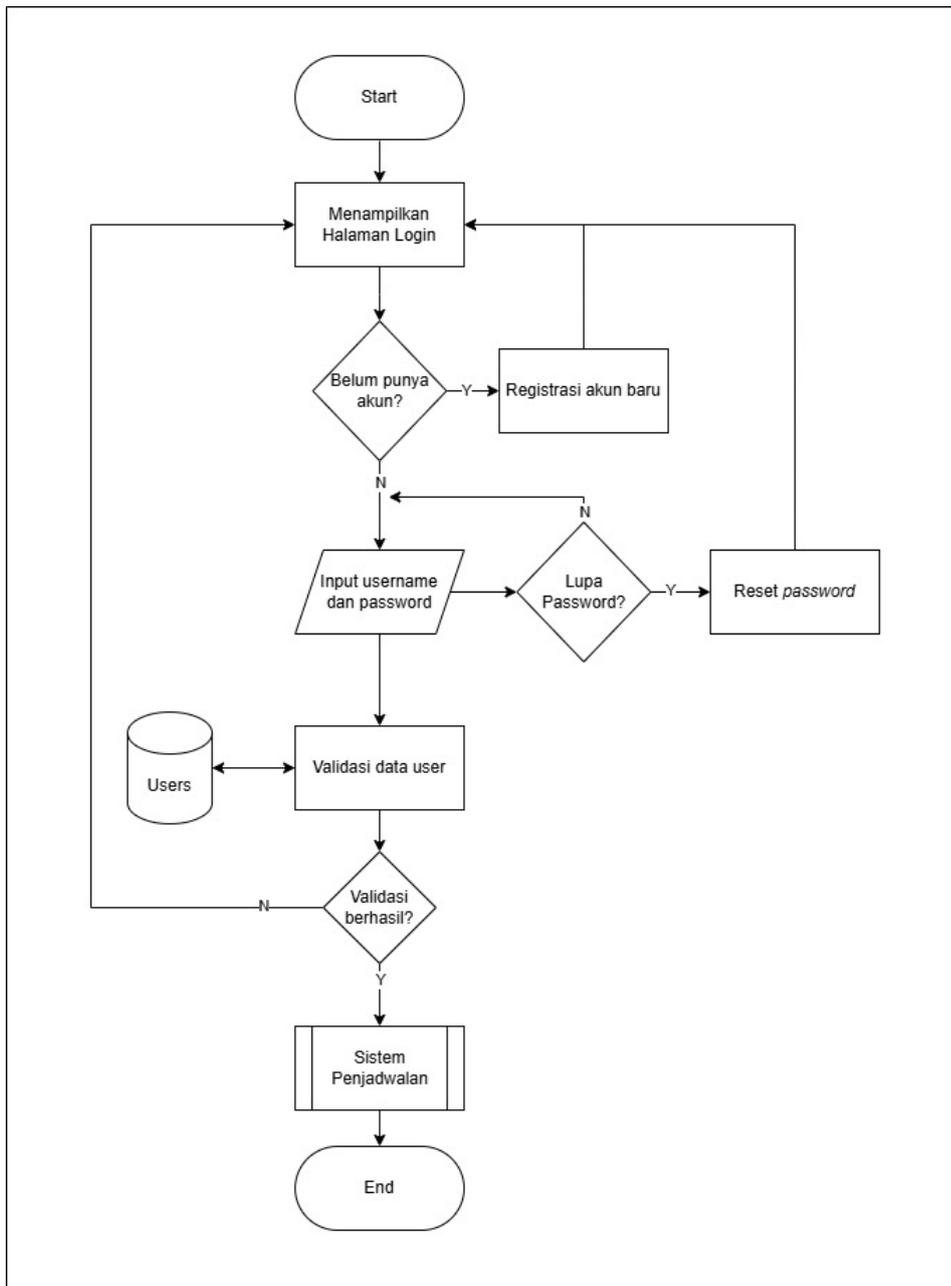
3.2 Perancangan Sistem

Pembangunan sistem ini dilakukan melalui beberapa tahap perancangan, yakni pembuatan *flowchart* untuk menggambarkan alur kerja sistem pada *website*, pembuatan *data flow diagram* (DFD) untuk menjabarkan alur data dalam sistem, pembuatan skema dan struktur *database* untuk mengetahui database yang digunakan, serta perancangan *interface website* berupa *mockup* untuk mendapatkan gambaran tampilan *website* dengan lebih jelas. Adapun pembuatan rancangan *flowchart*, DFD, dan skema *database* dilakukan menggunakan *software* Draw.io.

3.2.1 Diagram Flowchart

Perancangan sistem dimulai dengan membuat *flowchart* yang digunakan untuk menggambarkan alur kerja dari *website* penjadwalan aktivitas dan tugas perkuliahan. *Flowchart* ini dibuat dengan mengimplementasikan algoritma *Priority Scheduling*.

Sebelum dapat mengakses *website*, *user* diminta untuk melakukan login terlebih dahulu dengan memasukkan *username* dan *password* dari akun yang telah terdaftar. Jika *user* belum memiliki akun, maka *user* dapat melakukan registrasi untuk membuat akun baru sebelum melakukan login. *User* juga dapat melakukan *reset password* apabila menghadapi masalah dalam menginput *password*. Alur proses login dapat dilihat pada Gambar 3.1.

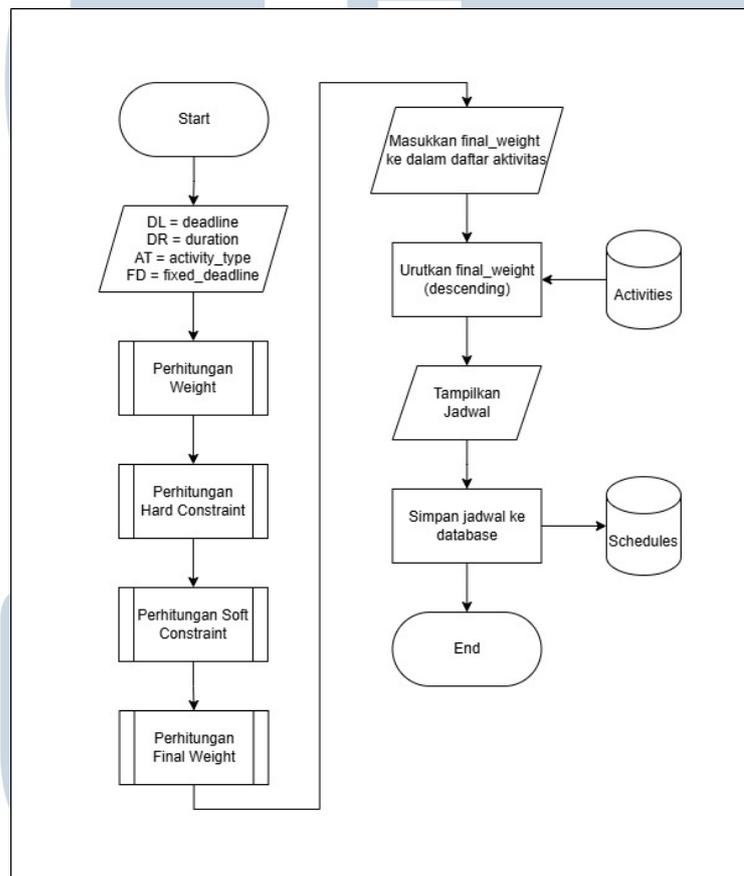


Gambar 3.1. Flowchart proses login

Gambar 3.2 adalah *flowchart* utama dari sistem yang dibuat untuk menggambarkan alur penggunaan *website*. Setelah akun berhasil divalidasi, *user* dapat mengakses halaman *Dashboard* yang merupakan *landing page* dari *website* setelah login. Dari *Dashboard*, *user* dapat mengakses halaman-halaman lain, seperti halaman *Input Activity*, halaman *Activity List*, halaman *Calculation Result*, halaman *Activity History*, dan juga halaman *Help*.

A Flowchart Proses Penjadwalan

Proses penjadwalan aktivitas dalam sistem ini menggunakan implementasi algoritma *Priority Scheduling* yang dirancang untuk mengatur jadwal dengan mempertimbangkan beberapa faktor prioritas, seperti tenggat waktu (*deadline*), durasi pengerjaan, dan jenis aktivitas, serta menggunakan batasan (*constraint*) untuk memaksimalkan efisiensi penjadwalan sistem. Adapun penjabaran alur proses penjadwalan tersebut dapat dilihat melalui Gambar 3.3.



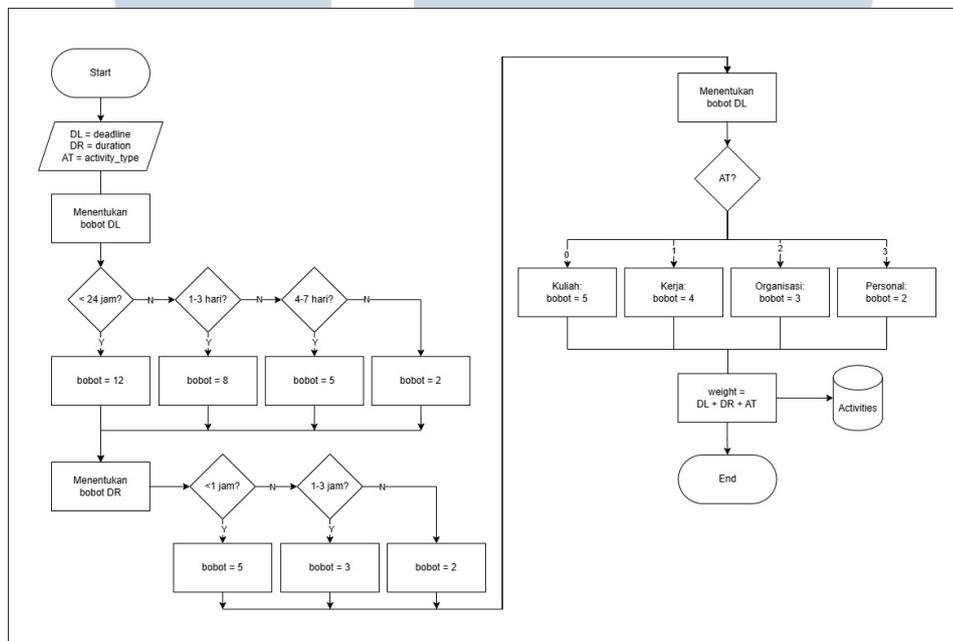
Gambar 3.3. Flowchart proses penjadwalan aktivitas

Proses kalkulasi penjadwalan aktivitas memiliki empat tahap proses perhitungan untuk menghasilkan perhitungan yang akurat. Proses perhitungan tersebut terdiri atas perhitungan bobot utama atau *weight*, perhitungan poin *Hard Constraint*, perhitungan poin *Soft Constraint*, dan perhitungan bobot akhir atau *final weight*. Seluruh aktivitas kemudian akan diurutkan berdasarkan hasil bobot akhir masing-masing secara *descending* lalu ditampilkan pada halaman *Calculation Result*. Untuk memastikan aktivitas dengan *deadline* mendekat diprioritaskan,

jadwal akan terus menghitung ulang $W_{deadline}$ setiap hari, menambahkan poin seiring tenggat waktu semakin dekat. Ini mendorong tugas dengan *deadline* mendesak untuk naik prioritasnya.

B Flowchart Perhitungan Weight

Gambar 3.4 menunjukkan proses perhitungan bobot awal. Setiap aktivitas yang diinput akan memiliki bobot (*weight*) berdasarkan tiga parameter utama: *deadline*, durasi pengerjaan, dan jenis aktivitas. Ketiga parameter ini akan menentukan bobot prioritas awal dari aktivitas sebelum adanya penyesuaian lebih lanjut.



Gambar 3.4. Flowchart perhitungan *weight*

Adapun penentuan bobot bagi setiap parameter (*deadline*, durasi, dan jenis aktivitas) dapat dijabarkan sebagai berikut:

1. *Deadline*

Jika waktu pengumpulan atau waktu tengat aktivitas mahasiswa semakin dekat, maka bobot dari aktivitas tersebut akan semakin tinggi.

- 1) Apabila batas waktu pengumpulan memiliki jangka waktu selama kurang dari 24 jam, maka bobot = 12.

- 2) Apabila batas waktu pengumpulan memiliki jangka waktu selama 1 sampai 3 hari, maka bobot = 8.
- 3) Apabila batas waktu pengumpulan memiliki jangka waktu selama 4 sampai 7 hari, maka bobot = 5.
- 4) Apabila batas waktu pengumpulan memiliki jangka waktu selama lebih dari 7 hari, maka bobot = 2

2. Durasi Pengerjaan

Aktivitas atau tugas yang memiliki waktu pengerjaan yang lebih pendek memiliki bobot lebih tinggi, dikarenakan lebih mudah untuk diselesaikan.

- 1) Apabila waktu yang dibutuhkan untuk menyelesaikan *task* memakan waktu kurang dari 1 jam, maka bobot = 5.
- 2) Apabila waktu yang dibutuhkan untuk menyelesaikan *task* memakan waktu 1 sampai 3 jam, maka bobot = 3.
- 3) Apabila waktu yang dibutuhkan untuk menyelesaikan *task* memakan waktu lebih dari 3 jam, maka bobot = 2.

3. Jenis Aktivitas

Setiap aktivitas akan dikategorikan ke dalam empat jenis aktivitas (kuliah, kerja, organisasi, dan personal) dimana setiap jenis aktivitas memiliki tingkat prioritas masing-masing.

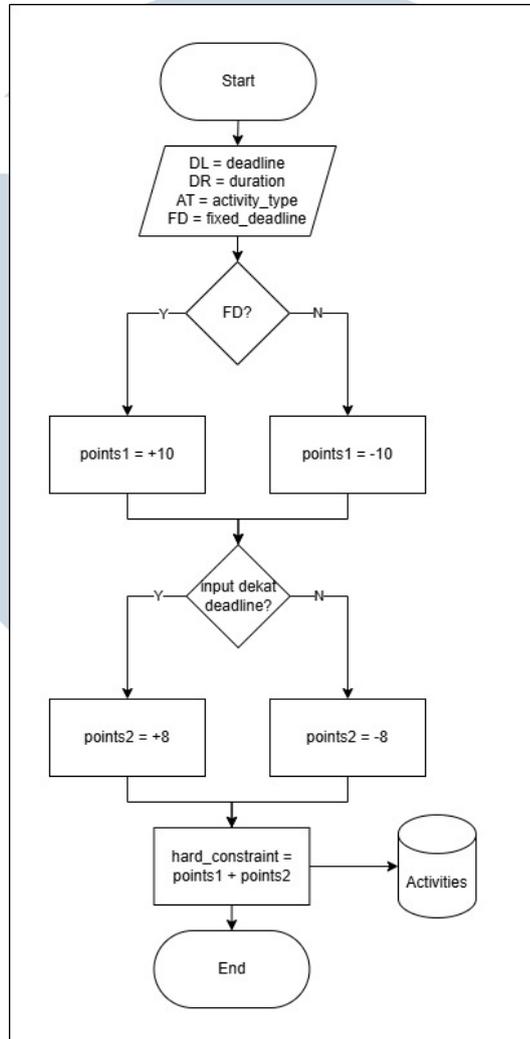
- 1) Apabila aktivitas termasuk ke dalam kategori Kuliah, maka bobot = 5.
- 2) Apabila aktivitas termasuk ke dalam kategori Kerja, maka bobot = 4.
- 3) Apabila aktivitas termasuk ke dalam kategori Organisasi, maka bobot = 3.
- 4) Apabila aktivitas termasuk ke dalam kategori Personal, maka bobot = 2.

Dengan bobot berdasarkan parameter-parameter ini, skor prioritas awal dapat dihitung untuk setiap aktivitas dengan menjumlahkan total bobot yang didapat dari parameter.

C Flowchart Perhitungan Constraint

Untuk meningkatkan akurasi perhitungan prioritas aktivitas dalam jadwal dan memperhitungkan kendala-kendala terdapat dalam dunia nyata, diterapkan

Hard Constraint dan *Soft Constraint* pada bobot awal untuk menyesuaikan prioritas aktivitas.



Gambar 3.5. Flowchart perhitungan *hard constraint*

Gambar 3.5 menunjukkan proses perhitungan poin *Hard Constraint*. *Hard Constraint* menerapkan persyaratan ketat, dimana kondisi pada batasan ini harus dipenuhi dan ketidakmampuan dalam memenuhi batasan ini akan mengakibatkan penalti yang menurunkan prioritas tugas. Terdapat dua *Hard Constraint*:

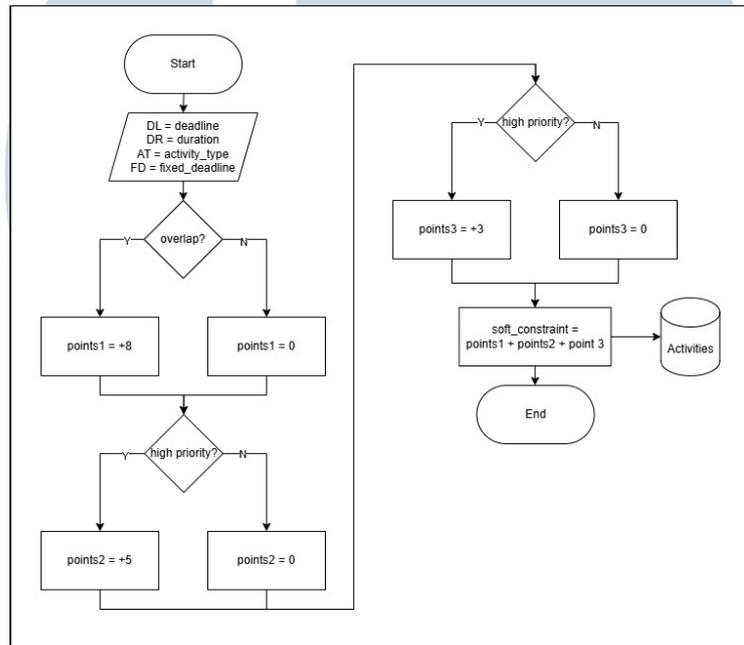
1. Fixed Deadline (Deadline Harus Terpenuhi):

- 1) Jika *deadline* harus terpenuhi, diberi tambahan +10 poin.
- 2) Jika *deadline* fleksibel, diberi penalti –10 poin.

2. Waktu Penjadwalan Aktivitas:

- 1) Jika aktivitas dijadwalkan dua hari sebelum tenggat waktunya, diberi tambahan +8 poin.
- 2) Jika tidak, diberi penalti –8 poin.

Poin dari batasan-batasan ini kemudian dijumlahkan untuk mendapatkan bobot *Hard Constraint*.



Gambar 3.6. Flowchart perhitungan *soft constraint*

Gambar 3.6 menunjukkan proses perhitungan poin *Soft Constraint*. *Soft Constraint* menerapkan kondisi yang ingin dipenuhi untuk meningkatkan efisiensi jadwal tanpa memberikan penalti apabila kondisi tersebut tidak terpenuhi. Terdapat dua *Soft Constraint*:

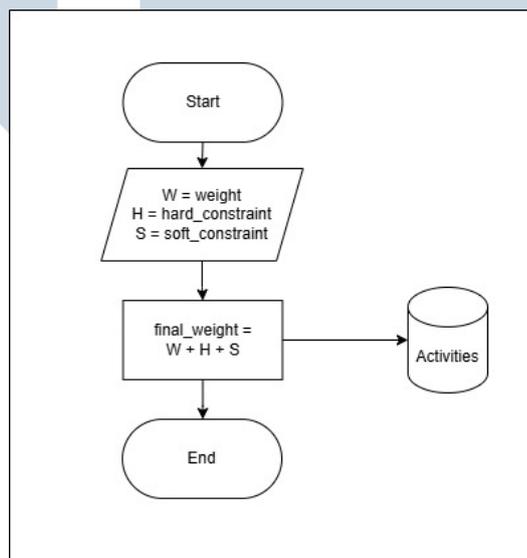
- 1) **Persyaratan tidak tumpang tindih:**
Menambahkan +8 poin jika aktivitas tidak tumpang tindih dengan aktivitas lain.
- 2) **Aktivitas prioritas tinggi:**
Menambahkan +5 poin jika aktivitas prioritas tinggi dijadwalkan dengan tepat.
- 3) **Distribusi aktivitas yang seimbang:**

Menambahkan +3 poin jika terdapat campuran yang seimbang dari jenis aktivitas,

Poin dari batasan-batasan ini kemudian dijumlahkan untuk mendapatkan bobot *Soft Constraint*.

D Flowchart Perhitungan Final Weight

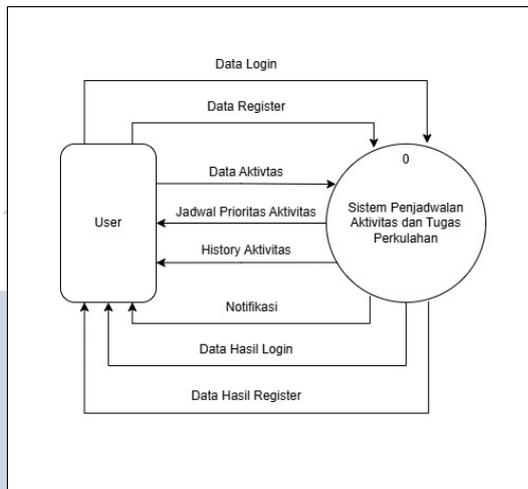
Gambar 3.7 menunjukkan proses perhitungan bobot akhir. Total Bobot Akhir W_{total} untuk setiap aktivitas dihitung dengan menggabungkan total bobot awal dengan batasan keras dan lunak, sehingga menghasilkan skor prioritas akhir. Total akhir pada bobot ini digunakan untuk menentukan urutan tugas yang akan dijadwalkan.



Gambar 3.7. Flowchart perhitungan *final weight*

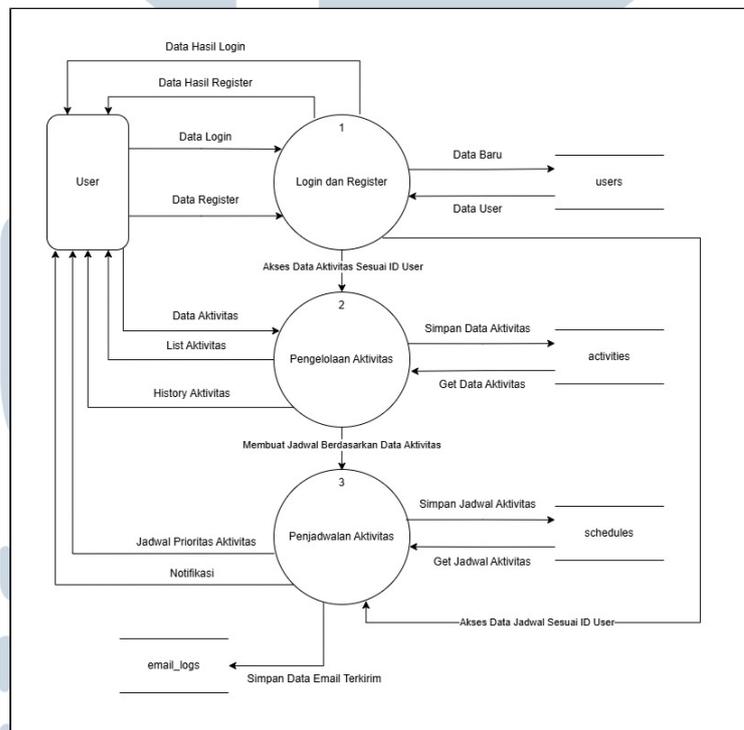
3.2.2 Data Flow Diagram

Gambar 3.8 menunjukkan *Data Flow Diagram* level 0 dari sistem penjadwalan. DFD level 0 ini menjelaskan alur jalannya sistem penjadwalan secara garis besar. Terdapat satu *entity* dalam sistem penjadwalan perkuliahan ini, yakni mahasiswa sebagai *user* dari sistem. Mahasiswa menginput aktivitas-aktivitas dan tugas perkuliahan ke dalam sistem yang kemudian akan menampilkan sebuah jadwal kegiatan berdasarkan prioritas kepada mahasiswa.



Gambar 3.8. *Data flow diagram level 0*

Gambar 3.9 menunjukkan *Data Flow Diagram* level 1 dari sistem penjadwalan yang menjelaskan alur sistem penjadwalan aktivitas perkuliahan ini dengan lebih detail.



Gambar 3.9. *Data flow diagram level 1*

Proses pada sistem penjadwalan penjadwalan terbagi menjadi tiga, yaitu proses "Login dan Register", "Pengelolaan Aktivitas", dan "Penjadwalan

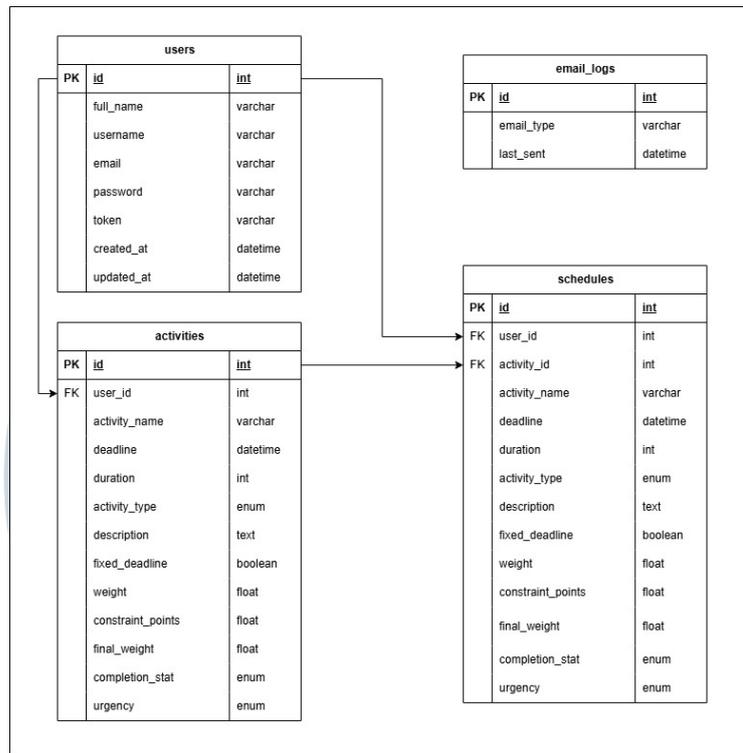
Aktivitas”. *User* melakukan login terlebih dahulu sebelum dapat mengakses *website*. Login dilakukan dengan memasukkan *username* serta *password* dari akun yang sebelumnya telah diregister dalam sistem. Apabila *user* belum memiliki akun, maka *user* dapat melakukan registrasi akun baru terlebih dahulu dengan mengisi formulir data pribadi serta *password* sebelum melakukan login. Setelah akun berhasil divalidasi, *user* dapat mengakses fitur-fitur yang terdapat dalam sistem.

Pengelolaan Aktivitas adalah proses dimana *user* dapat menginput aktivitas ke dalam sistem dengan mengisi formulir data aktivitas agar dapat disimpan ke dalam sistem. Dari data yang diinput, *user* dapat melihat daftar aktivitas yang terdapat dalam sistem serta melihat kembali *history* dari aktivitas-aktivitas yang sudah terselesaikan atau melampaui *deadline*. *User* kemudian juga dapat melakukan penjadwalan aktivitas dengan menggunakan data aktivitas yang telah terdaftar dalam sistem.

Penjadwalan Aktivitas merupakan fitur utama dalam sistem. Sistem akan memproses dan mengkalkulasi sebuah jadwal kegiatan yang dibuat dengan mengurutkan tingkat prioritas setiap aktivitas. Hasil dari kalkulasi ini kemudian akan ditampilkan kepada *user* dalam bentuk tabel jadwal aktivitas. Jadwal aktivitas ini bersifat *real-time* terus diperbaharui setiap hari, mengkalkulasi ulang prioritas setiap aktivitas yang berjalan mendekati *deadline*. Apabila *deadline* sudah dekat, maka sistem dapat memberikan notifikasi melalui email kepada *user* untuk mengingatkan akan aktivitas yang harus diselesaikan.

3.2.3 Skema Database

Gambar 3.10 di bawah ini menunjukkan skema database dari *database* sistem penjadwalan aktivitas perkuliahan mahasiswa. Tabel *users* berfungsi untuk menyimpan data pengguna atau mahasiswa. Tabel *activities* berfungsi untuk menyimpan data aktivitas yang telah diinput oleh mahasiswa ke dalam sistem.



Gambar 3.10. Skema database

3.2.4 Struktur Database

Berikut adalah struktur database dari tabel database yang digunakan dalam sistem penjadwalan.

1. Tabel *Users*

Tabel 3.1 menunjukkan struktur tabel *users* yang digunakan menyimpan data mahasiswa atau *user*.

Tabel 3.1. Struktur tabel *users*

Nama Kolom	Tipe Data	Key	Kesimpulan
<i>id</i>	INT	PK, <i>auto-increment</i>	ID pengguna
<i>full_name</i>	VARCHAR		Nama lengkap pengguna
<i>username</i>	VARCHAR	<i>Unique</i>	<i>Username</i> pengguna
<i>email</i>	VARCHAR	<i>Unique</i>	<i>Email</i> pengguna

Tabel 3.1 Struktur tabel *users* (lanjutan)

Nama Kolom	Tipe Data	Key	Kesimpulan
<i>password</i>	VARCHAR		<i>Password</i> akun pengguna
<i>token</i>	VARCHAR		<i>Token</i> untuk <i>reset password</i>
<i>created_at</i>	DATETIME		Tanggal dan waktu dibuat
<i>updated_at</i>	DATETIME		Tanggal dan waktu diperbaharui

Kolom-kolom pada tabel *user* terdiri atas 8 kolom, antara lain kolom *id user* yang digunakan untuk mengidentifikasi setiap *user* serta merupakan *primary key* dari tabel, kolom *full_name* digunakan untuk menyimpan nama lengkap *user*, kolom *username* digunakan untuk menyimpan *username* yang dapat digunakan untuk login, kolom *email* digunakan untuk menyimpan *email user*, kolom *password* digunakan untuk menyimpan *password*, kolom *token* untuk menyimpan *token* yang digunakan untuk melakukan *reset password*, kolom *created_at* digunakan untuk menyimpan tanggal serta waktu dibuatnya akun, dan kolom *updated_at* digunakan untuk menyimpan tanggal serta waktu diperbaharunya akun.

2. Tabel *Activities*

Tabel 3.2 menunjukkan struktur tabel *activities* yang digunakan untuk menyimpan data aktivitas perkuliahan atau tugas perkuliahan mahasiswa.

Tabel 3.2. Struktur tabel *activities*

Nama Kolom	Tipe Data	Key	Kesimpulan
<i>id</i>	INT	PK, <i>auto-increment</i>	ID aktivitas
<i>user_id</i>	INT	FK	Referensi ke tabel <i>Users</i>
<i>activity_name</i>	VARCHAR		Nama aktivitas atau tugas
<i>deadline</i>	DATETIME		<i>Deadline</i> aktivitas

Tabel 3.2 Struktur tabel *activities* (lanjutan)

Nama Kolom	Tipe Data	Key	Kesimpulan
<i>duration</i>	INT		Estimasi durasi penyelesaian aktivitas
<i>activity_type</i>	ENUM		Jenis aktivitas: <i>college, work, organization, personal</i>
<i>description</i>	TEXT		Deskripsi aktivitas
<i>weight</i>	FLOAT		Bobot aktivitas
<i>constraint_points</i>	FLOAT		Poin <i>hard constraint</i> dan <i>soft constraint</i>
<i>final_weight</i>	FLOAT		Total bobot akhir aktivitas
<i>fixed_deadline</i>	BOOLEAN		Penanda apakah <i>deadline</i> aktivitas bersifat tetap (<i>true</i>) atau fleksibel (<i>false</i>)
<i>completed_stat</i>	ENUM		Status penyelesaian aktivitas: <i>pending, overdue, completed</i>
<i>urgency</i>	ENUM		Tingkat urgensi aktivitas: <i>critical, high, medium, low</i>

Kolom-kolom pada tabel *activities* terdiri atas 12 kolom, yakni kolom id aktivitas yang digunakan untuk mengidentifikasi aktivitas serta merupakan *primary key* dari tabel, kolom *user_id* sebagai *foreign key* yang digunakan untuk merferensi pada tabel *users*, kolom *activity_name* digunakan untuk menyimpan nama aktivitas atau tugas perkuliahan, kolom *deadline*

digunakan untuk menyimpan waktu serta tanggal *deadline* dari aktivitas, kolom *duration* digunakan untuk menyimpan waktu estimasi pengerjaan aktivitas atau tugas, kolom *activity_type* digunakan untuk menyimpan kategori atau jenis aktivitas (aktivitas perkuliahan, kerja, organisasi, atau personal), kolom *description* menyimpan deskripsi tentang aktivitas, kolom *weight* digunakan untuk menyimpan total bobot dari aktivitas, kolom *constraint_points* digunakan untuk menyimpan total poin *constraint* aktivitas, kolom *final_weight* digunakan untuk menyimpan total bobot akhir dari aktivitas, kolom *fixed_deadline* digunakan sebagai penanda apakah *deadline* dari aktivitas bersifat tetap (*true*) atau fleksibel (*false*), kolom *completed_stat* digunakan untuk menentukan status penyelesaian aktivitas (masih dalam proses pengerjaan, melewati batas tanggal waktu, atau telah diselesaikan), dan kolom *urgency* digunakan untuk menentukan tingkat urgensi dari suatu aktivitas (kritis, tinggi, sedang, atau rendah).

3. Tabel *Schedules*

Tabel 3.3 menunjukkan struktur tabel *schedules*. Tabel *schedules* terdiri atas kolom yang sama dengan tabel *activities*. Hal yang membedakan kedua tabel ini, yaitu kolom *activity_id* pada tabel *schedules* yang berfungsi untuk mereferensi tabel *activities*. Adapun kegunaan pada tabel ini adalah untuk menyimpan data dari jadwal yang telah berhasil dibuat oleh sistem.

Tabel 3.3. Struktur tabel *schedules*

Nama Kolom	Tipe Data	Key	Kesimpulan
<i>id</i>	INT	PK, <i>auto-increment</i>	ID <i>schedule</i>
<i>user_id</i>	INT	FK	Referensi ke tabel <i>Users</i>
<i>activity_id</i>	INT	FK	Referensi ke tabel <i>Activities</i>
<i>activity_name</i>	VARCHAR		Nama aktivitas atau tugas
<i>deadline</i>	DATETIME		<i>Deadline</i> aktivitas

Tabel 3.3 Struktur tabel *schedules* (lanjutan)

Nama Kolom	Tipe Data	Key	Kesimpulan
<i>duration</i>	INT		Estimasi durasi penyelesaian aktivitas
<i>activity_type</i>	ENUM		Jenis aktivitas: <i>college, work, organization, personal</i>
<i>description</i>	TEXT		Deskripsi aktivitas
<i>weight</i>	FLOAT		Bobot aktivitas
<i>constraint_points</i>	FLOAT		Poin <i>hard constraint</i> dan <i>soft constraint</i>
<i>final_weight</i>	FLOAT		Total bobot akhir aktivitas
<i>fixed_deadline</i>	BOOLEAN		Penanda apakah <i>deadline</i> aktivitas bersifat tetap (<i>true</i>) atau fleksibel (<i>false</i>)
<i>completed_stat</i>	ENUM		Status penyelesaian aktivitas: <i>pending, overdue, completed</i>
<i>urgency</i>	ENUM		Tingkat urgensi aktivitas: <i>critical, high, medium, low</i>

4. Tabel *Email Logs*

Tabel 3.4 menunjukkan struktur tabel *email logs* yang untuk menyimpan data email yang telah dikirim oleh sistem kepada *user*. Tabel *email logs* terdiri atas tiga kolom: kolom *id* yang digunakan untuk mengidentifikasi log email serta merupakan *primary key* dari tabel, kolom *email_type* yang digunakan untuk

mengidentifikasi jenis email yang masuk ke dalam log email, dan kolom *last_sent* yang digunakan untuk mencatat waktu terkirimnya email.

Tabel 3.4. Struktur tabel *email logs*

Nama Kolom	Tipe Data	Key	Keterangan
<i>id</i>	INT	PK, auto-increment	ID <i>email</i>
<i>full_name</i>	VARCHAR		Nama lengkap pengguna
<i>email_type</i>	VARCHAR		Jenis <i>email</i>
<i>last_sent</i>	DATETIME		Waktu terakhir <i>email</i> dikirim

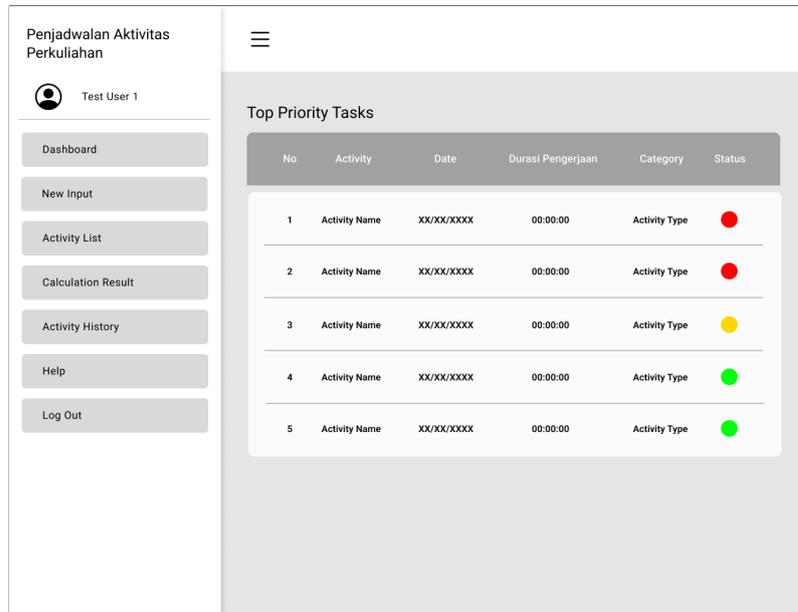
3.3 Desain Interface

Setelah sistem berhasil dirancang, maka dibuat sebuah desain *interface* atau disebut dengan *mockup* untuk merancang tampilan UI (*User Interface*) dari program. Berikut adalah rancangan *mockup* dari *website* Penjadwalan Aktivitas dan Tugas Perkuliahan.

3.3.1 Halaman Dashboard

Gambar 3.11 adalah rancangan halaman *Dashboard*. Halaman *Dashboard* merupakan halaman utama yang ditampilkan setelah *user* berhasil *login* menggunakan akun yang telah terdaftar. Halaman menunjukkan aktivitas yang sebaiknya diselesaikan terlebih dahulu dengan menampilkan aktivitas dengan tingkat prioritas paling tinggi.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.11. Rancangan halaman *dashboard*

3.3.2 Halaman Input Activity

Gambar 3.12 adalah rancangan halaman *Input Activity* dimana user dapat menginput aktivitas baru dengan mengisi formulir yang terdapat pada halaman. Formulir untuk menginput aktivitas baru terdiri atas kolom Aktivitas berupa *Text Box* untuk menginput nama aktivitas, kolom *Deadline* berupa input tanggal dan waktu tenggat dari aktivitas, kolom Estimasi Waktu Pengerjaan atau durasi untuk menginput perkiraan lama waktu penyelesaian aktivitas, kolom Jenis Aktivitas merupakan input *Dropdown* untuk mengkategorikan jenis aktivitas yang diinput, dan kolom *Fixed Deadline* berupa pilihan dalam bentuk input radio untuk menentukan apakah aktivitas yang diinput memiliki *deadline* yang wajib untuk ditemui atau bersifat fleksibel.

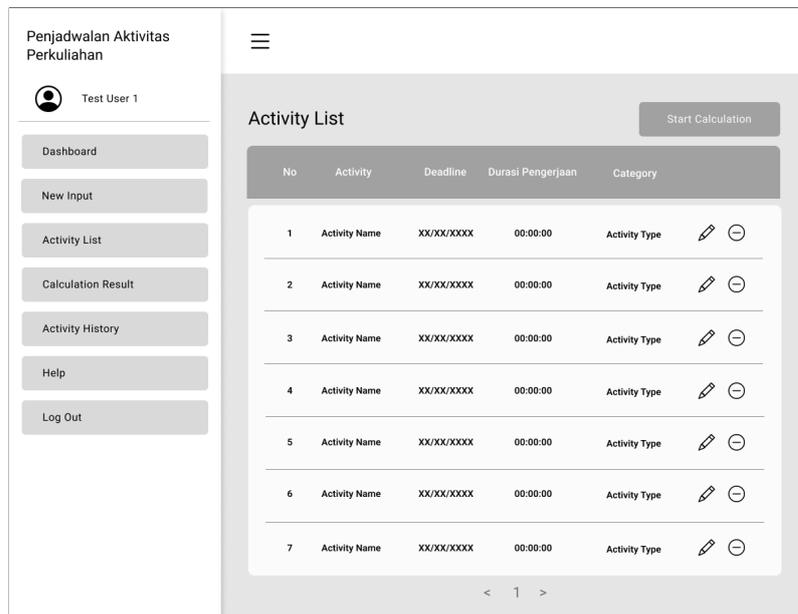
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Gambar 3.12. Rancangan halaman *input activity*

3.3.3 Halaman Activity List

Gambar 3.13 adalah rancangan halaman *Activity List* yang menampilkan aktivitas-aktivitas yang telah terdaftar dalam sistem. Aktivitas ditunjukkan dalam bentuk tabel yang berisi nomor, nama aktivitas, *deadline* aktivitas, durasi pengerjaan aktivitas, serta jenis aktivitas. *User* juga dapat mengedit dan menghapus aktivitas yang terdaftar dalam *list*. Halaman ini juga memiliki tombol *Start Calculation* yang berfungsi untuk memulai penjadwalan aktivitas pada sistem.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

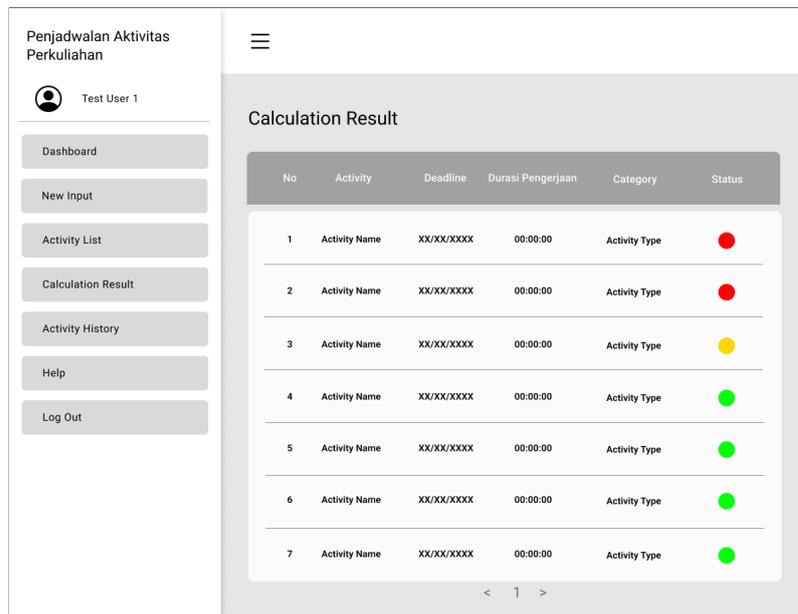


Gambar 3.13. Rancangan halaman *activity list*

3.3.4 Halaman Calculation Result

Gambar 3.14 adalah rancangan halaman *Calculation Result* yang menunjukkan hasil jadwal yang telah tersusun berdasarkan hasil kalkulasi prioritas aktivitas. Serupa dengan halaman *Activity List*, tampilan jadwal prioritas berisikan nama aktivitas, *deadline* aktivitas, durasi pengerjaan aktivitas, dan jenis aktivitas. Tampilan jadwal prioritas tidak memiliki tombol untuk mengedit atau menghapus aktivitas, melainkan terdapat tampilan status aktivitas yang terbagi menjadi tiga kategori: status warna hijau menandakan bahwa aktivitas tidak bersifat terlalu penting untuk dilaksanakan saat itu juga, status berwarna kuning menandakan bahwa aktivitas lebih baik untuk diselesaikan dalam waktu dekat, dan status berwarna merah menandakan bahwa aktivitas bersifat sangat *urgent* dan harus diutamakan atau diselesaikan sesegera mungkin.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

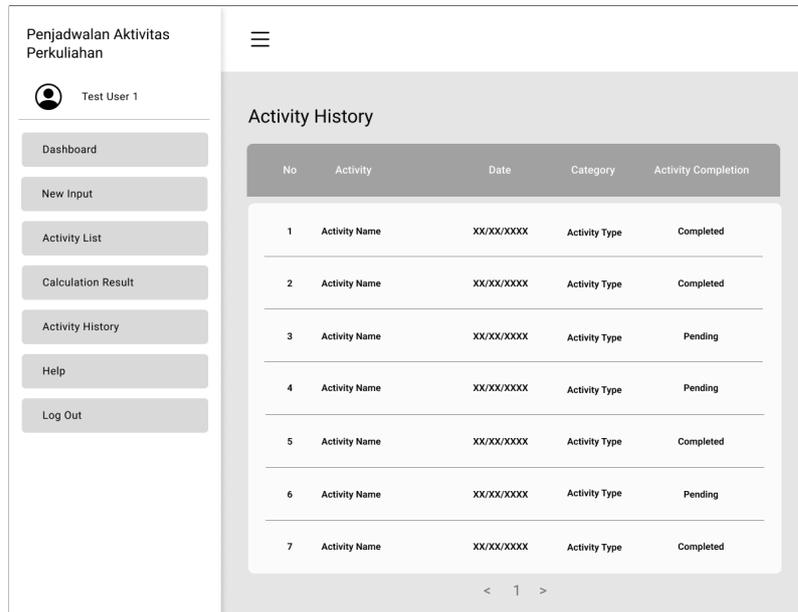


Gambar 3.14. Rancangan halaman *calculation result*

3.3.5 Halaman Activity History

Gambar 3.15 adalah rancangan halaman *Activity History* dimana halaman ini menampilkan daftar aktivitas yang sudah terselesaikan atau telah melewati *deadline*. Sejarah aktivitas ditampilkan dalam tabel yang berisi nama aktivitas, *deadline* aktivitas, jenis aktivitas, dan status penyelesaian aktivitas. Terdapat dua status penyelesaian aktivitas, yakni status *complete* yang menandakan bahwa aktivitas telah berhasil diselesaikan dan status *pending* yang menandakan bahwa aktivitas masih belum terselesaikan walau sudah melewati batas *deadline*.

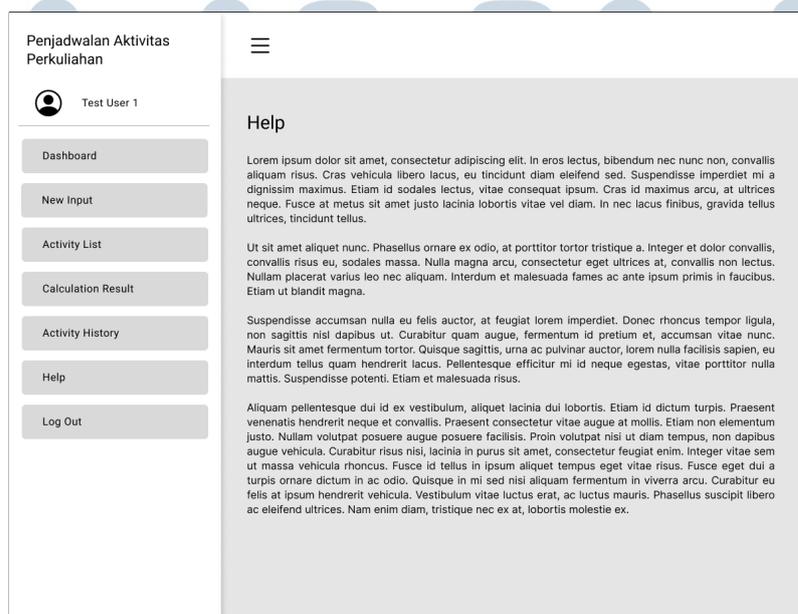
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.15. Rancangan halaman *activity history*

3.3.6 Halaman Help

Gambar 3.16 adalah rancangan halaman *Help* yang berisikan instruksi atau tata cara penggunaan website. Halaman *Help* ini memiliki tujuan untuk membantu *user* untuk menavigasi website beserta fitur-fiturnya dengan lebih mudah.



Gambar 3.16. Rancangan halaman *help*