

KERJA PRAKTEK
di
PT. Adiputro Wirasejati
Jl. Raya Balearjosari no. 35
Malang
14 Januari 2022 s/d 14 Juni 2022



Oleh:

Lawrence Patrick 219116837
Widean Nagari 219116863

PROGRAM SARJANA
PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
INSTITUT SAINS DAN TEKNOLOGI TERPADU SURABAYA
SURABAYA
2022

KERJA PRAKTEK
di
PT. Adiputro Wirasejati
Jl. Raya Balearjosari no. 35, Malang,
14 Januari 2022 s/d 14 Juni 2022”

Menyetujui
Pembimbing Lapangan

P.T. ADIPUTRO WIRASEJATI

M A L A N G

(Lydia Mirawati)

Mengetahui

Ketua Program Studi



(Dr. Yosi Kristian, S.Kom., M.Kom.)

Dosem Pembimbing



(Dr. Yosi Kristian, S.Kom., M.Kom.)

FAKULTAS SAINS DAN TEKNOLOGI
PROGRAM STUDI SARJANA INFORMATIKA
INSTITUT SAINS DAN TEKNOLOGI TERPADU SURABAYA
SURABAYA
2022

SURAT PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Lawrence Patrick
Fakultas/ Prodi : Sains dan Teknologi
NRP : 219116837

Nama : Widean Nagari
Fakultas/ Prodi : Sains dan Teknologi
NRP : 219116863

dengan ini menyatakan bahwa isi sebagian maupun keseluruhan Buku Kerja Praktek dengan judul:

**KERJA PRAKTEK di PT. Adiputro Wirasejati Jl. Raya Balearjosari no. 35, Malang,
14 Januari 2022 s/d 14 Juni 2022**

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 06-06-2021

Yang Membuat Pernyataan,



Lawrence Patrick
2191168637

Yang Membuat Pernyataan,



Widean Nagari
219116863

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa karena telah memberikan kesempatan pada penulis untuk menyelesaikan buku yang berjudul “Pengembangan Website Tes Psikologi PT. Adiputro Wirasejati” dengan tepat waktu.

Buku “Pengembangan Website Tes Psikologi PT. Adiputro Wirasejati” disusun guna memenuhi tugas Bapak Dr. Yosi Kristian, S.Kom., M.Kom. pada mata kuliah Kerja Praktek di ISTTS. Selain itu, penulis juga berharap agar makalah ini dapat menambah wawasan bagi pembaca tentang bagaimana fungsi dan fitur dari website Tes Psikologi PT. Adiputro Wirasejati. sehingga akhirnya dapat digunakan dengan baik sebagai sarana tes psikologi bagi para pelamar kerja di PT. Adiputro Wirasejati.

Penulis mengucapkan terima kasih sebesar-besarnya kepada Bapak Dr. Yosi Kristian, S.Kom., M.Kom selaku dosen mata kuliah Kerja Praktek dan Ibu Lydia Mirawati selaku Staf HRD PT. Adiputro Wirasejati. Tugas yang telah diberikan ini dapat menambah pengetahuan dan wawasan terkait pengembangan website tes psikologi.

Penulis menyadari buku ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun akan penulis terima demi kesempurnaan buku ini. Akhir kata, penulis menyampaikan terima kasih kepada semua pihak yang telah berperan serta dalam penyusunan buku ini dari awal sampai akhir. Semoga Tuhan senantiasa memberkati segala usaha kita. Amin.

Surabaya, 6 Juni 2022



Penulis

DAFTAR ISI

	halaman
SURAT PERNYATAAN KEASLIAN	iii
KATA PENGANTAR	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR	viii
DAFTAR TABEL.....	xi
DAFTAR ALGORITMA.....	xiii
BAB I PENDAHULUAN.....	15
1.1 Latar Belakang.....	15
1.2 Tujuan	16
1.3 Teori Penunjang.....	17
1.4 Garis Besar Sistem.....	18
1.5 Metodologi.....	19
1.6 Batasan Masalah	20
1.7 Penanggung Jawab.....	20
1.8 Gantt Chart.....	21
1.9 Sistematika Pembahasan.....	21
BAB II ANALISA SISTEM.....	23
2.1. Profil Perusahaan	23
2.2. Fact Finding	24
2.3. Deskripsi Sistem Baru.....	27
2.4. Kebutuhan Software.....	38
BAB III DESAIN SISTEM	41
3.1 Desain Arsitektur	41
3.2 Desain Database.....	41
3.2.1 Desain Entity Relationship Diagram	42
3.2.2 Desain Tabel	45
3.3 Desain User Interface.....	58
3.3.1 Halaman Job Applicant	58
3.3.2 Halaman Registrant	61

3.3.3 Halaman Admin.....	73
BAB IV IMPLEMENTASI	88
4.1 Kerangka Program	88
4.1.1 Backend-Clock	88
4.1.2 Backend	89
4.1.3 Frontend.....	93
4.2 Implementasi Program.....	96
4.2.1 Backend-Clock	96
4.2.2 Backend	99
4.2.3 Frontend.....	110
BAB V UJI COBA DAN DEPLOYMENT	139
5.1 Uji Coba.....	139
5.1.1 Login.....	139
5.1.2 Rekrutmen	140
5.1.3 Halaman Registrant	140
5.1.4 Halaman Admin.....	143
5.1.4.1 Master Registrant.....	143
5.1.4.2 Master Tes	144
5.1.4.3 Master Exam Session.....	147
5.1.4.4 Master Recruitment	149
5.1.5 Kendala dalam Testing.....	151
5.2 Deployment.....	152
5.2.1 Proses Deployment.....	152
5.2.2 Kendala Dalam Deployment	154
BAB VI PENUTUP	156
6.1 Kesimpulan	156
6.2 Saran	156
BAB VII USER MANUAL	158
7.1 Halaman Login.....	158
7.2 Halaman Dashboard Registrant	159
7.3 Halaman Daftar Persoalan	160

7.4	Halaman Pre-Test.....	161
7.5	Halaman Tes Pilihan Ganda.....	162
7.6	Tampilan Daftar Soal.....	163
7.7	Halaman Tes gambar	163
7.8	Halaman Form Biodata.....	164
7.9	Halaman Dashboard Admin.....	166
7.10	Halaman Master Registrant.....	167
7.11	Halaman Detail Registrant.....	168
7.12	Halaman Hasil Biodata	169
7.13	Halaman Master Tes	170
7.14	Komponen Buat Seksi Baru.....	171
7.15	Komponen Pertanyaan Baru	172
7.16	Komponen Perbarui Pertanyaan	172
7.17	Halaman Master Sesi	173
7.18	Komponen Sesi Baru dan Perbarui Sesi	174
7.19	Halaman Master Rekrutmen	176
7.20	Komponen Rekrutmen Baru	177

DAFTAR GAMBAR

	halaman
Gambar 1. 1 Waterfall Workflow	20
Gambar 1. 2 Gantt Chart.....	21
Gambar 2. 1 Lembar Jawaban Tes Adkudag Persoalan 1	25
Gambar 2. 2 Lembar Jawaban Tes Papi Kostick	26
Gambar 2. 3 Lembar Jawaban Tes Ekspresi	27
Gambar 2. 4 Use Case User	28
Gambar 2. 5 Use Case Job Applicant	29
Gambar 2. 6 Use Case User Biasa	29
Gambar 2. 7 Use Case Admin.....	30
Gambar 2. 8 Activity Diagram Job Applicant – Apply Pekerjaan	31
Gambar 2. 9 Activity Diagram Registrant – Kerja Tes	32
Gambar 2. 10 Activity Diagram Registrant – Isi Biodata.....	33
Gambar 2. 11 Activity Diagram Admin – Master Registrant.....	34
Gambar 2. 12 Activity Diagram Admin – Master Test	35
Gambar 2. 13 Activity Diagram Admin – Update Session.....	36
Gambar 2. 14 Activity Diagram Admin – Master Recruitment.....	38
Gambar 3. 1 Arsitektur Sistem.....	41
Gambar 3. 2 ERD Diagram (PDM)	44
Gambar 3. 3 Halaman Login.....	58
Gambar 3. 4 Halaman Apply Pekerjaan	60
Gambar 3. 5 Halaman Dashboard Registrant	61
Gambar 3. 6 Halaman Dashboard Persoalan	62
Gambar 3. 7 Halaman Pre-Test.....	63
Gambar 3. 8 Halaman Tes Biasa.....	65
Gambar 3. 9 Tampilan Daftar Nomor Soal.....	66
Gambar 3. 10 Tampilan Pertanyaan Gambar.....	66
Gambar 3. 11 Tampilan Jawaban Isian.....	66
Gambar 3. 12 Tampilan Jawaban Gambar.....	67

Gambar 3. 13 Halaman Tes EPPS/Papi Kostick.....	68
Gambar 3. 14 Halaman Tes Kraepelin.....	68
Gambar 3. 15 Halaman Tes Ekspresi.....	70
Gambar 3. 16 Halaman Tes Gambar.....	71
Gambar 3. 17 (a) Form Biodata (atas) (b) Form Biodata (bawah)	72
Gambar 3. 18 Tampilan Header Admin.....	73
Gambar 3. 19 Navigation Bar Admin.....	74
Gambar 3. 20 Halaman Dashboard Admin.....	75
Gambar 3. 21 Halaman Master Registrant.....	76
Gambar 3. 22 Halaman Detail Registrant (atas)	77
Gambar 3. 23 Halaman Detail Registrant (bawah).....	78
Gambar 3. 24 Halaman Hasil Biodata	78
Gambar 3. 25 Halaman Master Sesi	79
Gambar 3. 26 (a) Modal Sesi Baru (b) Modal Perbarui Sesi	80
Gambar 3. 27 Halaman Master Tes (atas)	81
Gambar 3. 28 Halaman Master Tes (bawah)	82
Gambar 3. 29 Tampilan Modal Seksi Baru	83
Gambar 3. 30 Tampilan Modal Pertanyaan Baru	84
Gambar 3. 31 Tampilan Modal Update Pertanyaan.....	84
Gambar 3. 32 (a) Master Rekrutmen (atas) (b) Master Rekrutmen (bawah).....	85
Gambar 3. 33 Tampilan Modal Rekrutmen Baru	86
Gambar 3. 34 Halaman Detail Pelamar	87
Gambar 4. 1 Struktur Folder Backend-Clock	88
Gambar 4. 2 Struktur Folder Backend	90
Gambar 4. 3 Struktur Folder Frontend.....	93
Gambar 7. 1 Form Login	158
Gambar 7. 2 Tampilan Halaman Dashboard Registrant	159
Gambar 7. 3 Tampilan Halaman Daftar Persoalan	160
Gambar 7. 4 Tampilan Halaman Pre-Test	161
Gambar 7. 5 Tampilan Halaman Tes Pilihan Ganda	162
Gambar 7. 6 Tampilan Daftar Soal	163

Gambar 7. 7 Tampilan Tes Gambar.....	164
Gambar 7. 9 Tampilan Halaman Dashboard Admin	166
Gambar 7. 10 Tampilan Halaman Master Registrant	167
Gambar 7. 11 Tampilan Halaman Detail Registrant.....	168
Gambar 7. 12 Tampilan Halaman Hasil Biodata.....	169
Gambar 7. 13 Tampilan Halaman Master Tes.....	170
Gambar 7. 14 Tampilan Komponen Buat Seksi Baru.....	171
Gambar 7. 15 Tampilan Komponen Pertanyaan Baru	172
Gambar 7. 16 Tampilan Komponen Perbarui Pertanyaan	173
Gambar 7. 17 Tampilan Halaman Master Sesi	174
Gambar 7. 18 (a) Komponen Sesi Baru (b) Komponen Perbarui Sesi	175
Gambar 7. 19 Tampilan Halaman Master Rekrutmen.....	176
Gambar 7. 20 Tampilan Komponen Rekrutmen Baru.....	177

DAFTAR TABEL

	Halaman
Tabel 3. 1 Tabel “Job_Vacancy”	45
Tabel 3. 2 Tabel “Applicant”	46
Tabel 3. 3 Tabel “Exam_Session”	47
Tabel 3. 4 Tabel “Registrant”	48
Tabel 3. 5 Tabel “Test”	49
Tabel 3. 6 Tabel “Exam_Session_Test”	49
Tabel 3. 7 Tabel “Section”	50
Tabel 3. 8 Tabel “Question”	51
Tabel 3. 9 Tabel “Test_Result”	52
Tabel 3. 10 Tabel “Section_Result”	53
Tabel 3. 11 Tabel “Question_Result”	54
Tabel 3. 12 Tabel “IST_Data”	54
Tabel 3. 13 Tabel “Kreapelin_Data”	55
Tabel 3. 14 Tabel “EPPS_Data”	56
Tabel 3. 15 Tabel “Section_Ongoing”	57
Tabel 4. 1 Tabel Folder Backend/Controller	90
Tabel 4. 2 Tabel Folder Backend/Controller/Test_Result_Calc	91
Tabel 4. 3 Tabel Folder Backend/Helpers	91
Tabel 4. 4 Tabel Folder Backend/Models	91
Tabel 4. 5	92
Tabel Folder Backend/Routers/Express_Routers	92
Tabel 4. 6 Tabel Folder Backend/Setup	92
Tabel 4. 7 Tabel Folder Frontend/Assets	94
Tabel 4. 8 Tabel File Frontend/Components	94
Tabel 4. 9 Tabel Folder Frontend/Components/Report	94
Tabel 4. 10 Tabel Folder Frontend/Components/Views	94
Tabel 4. 11 Tabel Folder Frontend/Views	95

Tabel 4. 12 Tabel Folder Frontend/Views/Admin.....	95
Tabel 5. 1 Tabel Uji Coba Login	139
Tabel 5. 2 Tabel Uji Coba Rekrutmen.....	140
Tabel 5. 3 Tabel Uji Coba Isi Biodata	141
Tabel 5. 4 Tabel Uji Coba Mengerjakan Tes.....	141
Tabel 5. 5 Tabel Uji Coba Upload Kertas Buram.....	142
Tabel 5. 6 Tabel Uji Coba Layar Responsif	143
Tabel 5. 7 Tabel Uji Coba Cari Registrant	144
Tabel 5. 8 Tabel Uji Coba Cari Registrant	144
Tabel 5. 9 Tabel Uji Coba Tambah Persoalan.....	145
Tabel 5. 10 Tabel Uji Coba Update Persoalan.....	145
Tabel 5. 11 Tabel Uji Coba Tambah Pertanyaan.....	146
Tabel 5. 12 Tabel Uji Coba Update Pertanyaan	147
Tabel 5. 13 Tabel Uji Coba Cari Sesi	148
Tabel 5. 14 Tabel Uji Coba Tambah Sesi.....	148
Tabel 5. 15 Tabel Uji Coba Update Sesi.....	149
Tabel 5. 16 Tabel Uji Coba Tambah Lowongan Kerja.....	150
Tabel 5. 17 Tabel Uji Coba Update Lowongan Kerja	150
Tabel 5. 18 Tabel Uji Coba Detail Pelamar.....	151

DAFTAR ALGORITMA

	halaman
Segmen Program 4.1 Sequelize	96
Segmen Program 4.2 Database Model.....	97
Segmen Program 4.3 Clock Tick	98
Segmen Program 4.4 Database Model.....	100
Segmen Program 4.5 Insert ke Database	101
Segmen Program 4.6 Update ke Database	102
Segmen Program 4.7 Delete dari Database.....	103
Segmen Program 4.8 Mengambil data dari Database.....	104
Segmen Program 4.9 Upload ke Google Drive	105
Segmen Program 4.10 Perhitungan Norma Tes.....	107
Segmen Program 4.11 Socket.....	108
Segmen Program 4.12 Mailer	109
Segmen Program 4.13 Login	110
Segmen Program 4.14 Membuat PDF	112
Segmen Program 4.15 Ganti Data Kraepelin.....	114
Segmen Program 4.16 Reset Hasil Jawaban.....	115
Segmen Program 4.17 Membuka Modal Update Sesi	116
Segmen Program 4.18 Membuat dan Memperbarui Sesi	118
Segmen Program 4.19 Membuat Pertanyaan.....	121
Segmen Program 4.20 Memperbarui Pertanyaan	123
Segmen Program 4.21 Mereset Pertanyaan	125
Segmen Program 4.22 Logout	126
Segmen Program 4.23 Mencari Section/Persoalan Yang Sudah Diselesaikan.....	127
Segmen Program 4.24 Submit Data IST.....	129
Segmen Program 4.25 Pindah Ke Soal Berikutnya	131
Segmen Program 4.26 Mengirimkan hasil jawaban	132
Segmen Program 4.27 Mengupload jawaban saat ini.....	134
Segmen Program 4.28 Mengambil jawaban tes saat ini	135

Segmen Program 4.29 Inisialisasi Data Lama	136
Segmen Program 4.30 Submit Biodata.....	137

BAB I

PENDAHULUAN

Tes psikologi atau psikotes adalah bidang yang ditandai dengan penggunaan sampel perilaku untuk menilai konstruksi psikologis, seperti fungsi kognitif dan emosional, tentang individu tertentu. Website tes psikologi adalah sarana pelaksanaan tes psikologi yang berbasis online sehingga dapat dilakukan dimanapun dan kapanpun selama peserta memiliki akses internet dan masih memiliki akses ke tes tersebut. Pada bab ini, dibahas latar belakang dan alasan pembuatan website, tujuan pembuatan website, teori-teori yang menunjang pembuatan website, arsitektur program, dan penjelasan cara berjalannya proyek ini dimulai dari metodologi yang dipakai, penanggung jawab tiap bagian, sampai gantt chart yang menjadi pedoman/acuan tugas yang akan dikerjakan oleh tiap anggota tim.

1.1 Latar Belakang

Pada suatu perusahaan, pegawai adalah hal yang penting dan diperlukan karena merupakan penggerak dari perusahaan itu sendiri. Pentingnya karyawan tersebut membuat perusahaan harus memperhatikan kualitas dari setiap pegawainya agar tidak muncul masalah-masalah yang berarti. Salah satu cara untuk mengetahui kualitas tiap pegawai maupun calon pegawai adalah melalui tes. Tidak hanya tes secara hard skill, pegawai juga perlu dicek psikologinya karena kondisi mental seorang karyawan juga mempengaruhi kinerja karyawan di perusahaan.

Tes Psikologi biasanya dilakukan oleh calon pegawai yang ingin masuk bergabung ke dalam suatu perusahaan. Metode tes psikologi yang dilakukan biasanya on-paper atau tertulis di kertas. Dikarenakan pandemi covid-19 yang tak kunjung berhenti dan munculnya berbagai varian virus baru, alternatif tes dapat dilakukan secara daring atau online. Tes Online secara daring ini pun dapat dilakukan dari berbagai perangkat seperti komputer, laptop, ataupun smartphone.

Dalam menjaga kualitas produk yang dihasilkan PT Adiputro Wirasejati, tentunya mereka harus memperhatikan kualitas dari pegawainya. Maka dari itu tes masuk Adi Putro pun diperketat dengan adanya tes psikologi. Hal ini pun yang ingin diimplementasikan oleh

PT Adiputro Wirasejati. Penerimaan calon pegawai di perusahaan Adiputro harus melalui beberapa tes termasuk tes psikologi. Dikarenakan beberapa pegawai yang dapat berasal dari berbagai penjuru Indonesia dan adanya pandemi, maka dibutuhkannya tes psikotes secara online ini.

Untuk mengotomatisasi proses tes psikologi pada PT Adiputro Wirasejati, dirasa perlu membangun website yang dapat digunakan untuk pelaksanaan tes psikologi online, termasuk penggerjaan soal, penghitungan hasil tes otomatis, reporting, dan juga fitur master yang dapat digunakan untuk mengolah semua data yang ada pada database yang digunakan yang sangat diperlukan oleh pemilik perusahaan PT Adiputro Wirasejati. Dengan demikian diharapkan website ini dapat mengotomatisasi proses tes psikologi, dimana hasil tes akan selalu valid dan tepat serta laporan tes dapat dibuat secara otomatis.

Dengan kata lain solusi yang ditawarkan adalah membuat website tes psikologi yang dapat digunakan oleh :

- a. Calon pegawai perusahaan PT Adiputro Wirasejati, untuk melakukan tes psikologi.
- b. Staf bagian Human Resources Department (HRD) Perusahaan PT Adiputro Wirasejati, untuk melihat laporan hasil tes calon pegawai dan mengolah data pada database.

Aplikasi yang dibuat merupakan aplikasi web dan dapat digunakan dimanapun asalkan pengguna memiliki akses ke internet.

1.2 Tujuan

Dalam pembuatan aplikasi website dalam kerja praktek ini, terdapat beberapa tujuan yang ingin dicapai penulis, antara lain:

1. Mengembangkan website tes psikologi untuk PT. Adiputro Wirasejati
2. Mengembangkan fitur rekrutmen agar website yang dibuat dapat sekaligus digunakan untuk kepentingan rekrutmen pegawai baru.
3. Mengembangkan fitur tampilan website yang responsif agar tes psikologi dapat dilakukan menggunakan laptop, PC, tablet, maupun smartphone
4. Menguji coba semua alat tes yang dikembangkan

5. Memastikan website dapat dijalankan dengan lancar pada server milik PT. Adiputro Wirasejati

1.3 Teori Penunjang

Dalam penggeraan proyek ini, terdapat teori-teori yang menunjang mendukung pengembangan proyek ini yaitu :

A. Front End

Frontend merupakan bagian dari sebuah aplikasi yang dapat dilihat secara langsung oleh user, dan user dapat melakukan interaksi dengan bagian ini. Pada kali ini bagian frontend akan dibuat menggunakan HTML versi 5. HTML adalah sebuah markup language yang digunakan untuk menampilkan konten yang diinginkan ke browser. Untuk mempercantik halaman website, dibutuhkan CSS atau Cascading Style Sheet. Terdapat banyak cara dalam menulis CSS, dapat secara manual atau dapat menggunakan framework CSS. Dalam penggeraan proyek ini, akan digunakan framework CSS bernama Tailwind. Penggunaan Tailwind sangatlah mudah dimengerti dan diganti sesuka hati kita. Tidak seperti framework CSS lainnya, Tailwind memiliki keunggulan yaitu untuk menggunakan gabungan beberapa class milik Tailwind untuk bisa dikustomisasi. Selain itu, Tailwind juga memiliki ukuran yang relatif sangat kecil karena ia hanya meng-compile style yang digunakan saja. Tailwind yang akan digunakan adalah versi 2.0.

Proyek ini juga akan menggunakan framework Vue. Framework Vue adalah salah satu library Javascript yang digunakan untuk membantu dalam pengembangan interface web. Keuntungan dari Vue adalah kemudahan dalam pembuatan SPA (Single Page Application). Program yang ditulis dengan Vue biasanya memiliki file dengan ekstensi “.vue”. Versi Vue yang digunakan adalah versi 2.5.16. Proyek ini juga akan menggunakan jQuery, salah satu library milik Javascript yang membantu dalam penulisan script javascript. Versi Javascript yang digunakan adalah versi ECMAScript 5 dan Jquery yang digunakan adalah versi 3.3.

B. Back End

Backend merupakan bagian dari aplikasi yang berada di sisi belakang website atau dengan kata lain tidak dapat dilihat langsung oleh user. Pada bagian ini user tidak dapat langsung melihat proses penggeraan backend karena akan dikerjakan pada server side. Pada website ini, backend akan menggunakan Express.js. Express.js adalah salah satu library

Javascript dari NodeJS yang dirancang secara flexibel dan minimalis dimana Express.js ini dapat digunakan untuk membantu dalam pembuatan web application dan juga API. Keuntungan dari Express adalah kemudahan dalam membuat routing API. Express.js memberikan kebebasan bagi para developer untuk mendesain aplikasi, sehingga memungkinkan bagi setiap pengembang memiliki rancangan arsitektur yang berbeda dalam *software* yang dibangun. Express.js didukung . Selain itu, Express.Js juga didukung oleh mesin Google V8, sehingga membuat kinerja dari aplikasi lebih tinggi. Versi Express.js yang dipakai adalah versi 4.17.2.

C. Database

Database merupakan kumpulan informasi yang disimpan dalam komputer secara sistematis. Terdapat banyak cara untuk menyimpan informasi dalam database, salah satunya adalah dengan metode Relational Database, yang biasa disebut dengan Relational Database Management System. Pada proyek ini, kami menggunakan MySQL sebagai databasenya. MySQL merupakan salah satu Open Source Relational SQL Database Management System yang paling populer. Oleh karena itu kami memilih MySQL sebagai database web kami.

1.4 Garis Besar Sistem

Website tes psikologi ini memiliki sebuah sistem. Desain sistem yang dibuat akan dijelaskan lebih lanjut pada bab 3. Berikut adalah garis besar dari sistem website tes psikologi yang akan dibuat:

3.2.1 User

Dalam website ini terdapat 3 jenis user, yaitu User biasa, Moderator, dan Admin. Setiap user memiliki peran masing-masing, sebagai berikut:

1. Job Applicant : user ini berperan sebagai pelamar kerja.
2. User Biasa : user ini akan berperan sebagai peserta tes psikologi.
3. Admin : admin berperan sebagai pengelola master untuk tiap data di dalam website.

Tiap jenis user memiliki fitur website masing-masing. Fitur yang dapat digunakan oleh tiap jenis user dapat dilihat pada subbab Fitur Program.

3.2.2 Fitur Program

Dalam website ini akan terdapat banyak fitur yang disediakan untuk mendukung kenyamanan user dalam menggunakan website. Website ini juga memiliki role dimana yang

dapat menggunakan fitur ini hanya yang login dengan role yang sudah ditentukan. Fitur-fitur akan dijelaskan dalam bentuk tabel setiap user, nama fitur dan penjelasan singkat. Berikut fitur-fitur yang terdapat pada website ini:

A. Fitur Job Applicant

Pada website ini, job applicant dapat melamar kerja dengan cara scan barcode yang ada. Setelah diarahkan ke halaman pengisian data diri, pelamar dapat mengisikan data diri dan mengupload dokumen yang diminta (Contoh: CV, Surat Lamaran, Portofolio, dll.).

B. Fitur Registrant

Pada website ini, Registrant dapat Login menggunakan email saat mendaftar dan test token yang akan diberikan melalui email. Di dalam website, Registrant dapat mengerjakan semua tes yang telah diberikan. Registrant juga dapat mengisi biodata.

C. Fitur Admin

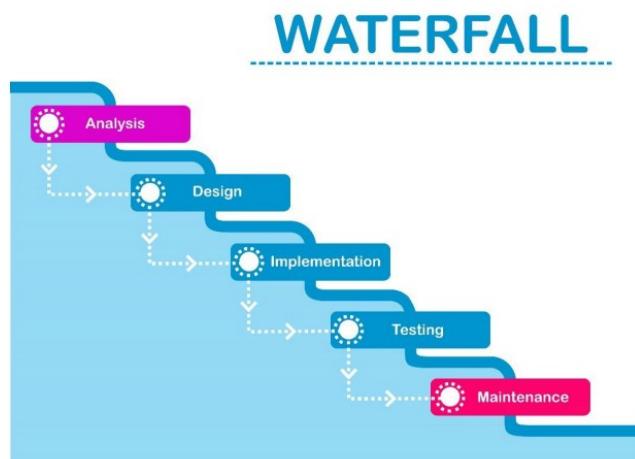
Pada website ini, admin dapat mengakses semua data yang ada mulai dari daftar user, tes, persoalan, pertanyaan, hasil tes, sesi, hingga data rekrutmen. Admin dapat menambah, mengupdate, dan menghapus data yang ada. Admin dapat memilih tes mana saja yang akan diberikan kepada registrant untuk dikerjakan. Selain itu, admin juga dapat melihat dan mengunduh laporan hasil tes registrant. Untuk kepentingan rekrutmen, admin dapat menambahkan lowongan pekerjaan dan melihat data dari pelamar kerja dari lowongan tersebut. Lowongan pekerjaan yang telah ditambahkan juga dapat ditutup sehingga orang tidak dapat mendaftar lagi.

1.5 Metodologi

Dalam pengembangan website ini, tentunya kita menggunakan metodologi pengembangan perangkat lunak (Software Development Methodologies). Metodologi yang digunakan adalah Waterfall. Waterfall Development Methods adalah salah satu metodologi pengembangan perangkat lunak yang sangat sederhana dan populer dipakai. Metodologi ini memanfaatkan proses berurutan dan sistematis dalam fase-fase penggerjaan software. Model pengembangannya dapat dianalogikan seperti air terjun, dimana setiap tahap dikerjakan secara berurutan mulai dari atas hingga ke bawah.

Intisari dari Waterfall Development adalah agar alur penggerjaan yang jelas dan terukur. Terdapat alokasi waktu yang pasti agar setiap tim dapat menyelesaikan pekerjaan

tepatis waktu. Selain dari itu, Waterfall juga menghemat biaya kepada klien dikarenakan tidak perlu membayar biaya tambahan pada saat ingin memberikan masukan atau feedback ke tim pengembang seperti pada metodologi Agile. Pembuatan aplikasi berskala besar juga cocok menggunakan waterfall.



Gambar 1. 1
Waterfall Workflow

1.6 Batasan Masalah

Dalam pembuatan website tes psikologi ini terdapat batasan-batasan sebagai berikut:

1. Tes tidak dapat dikerjakan tanpa koneksi internet
2. Format pembuatan tes tidak dapat dikustomisasi sesuai kebutuhan sehingga jika ingin membuat format tes baru harus menambah sistem baru
3. Tampilan responsive hanya bekerja pada halaman peserta tes. Halaman admin tidak dilengkapi dengan tampilan responsive
4. Website hanya bekerja optimal pada browser Microsoft Edge dan Google Chrome

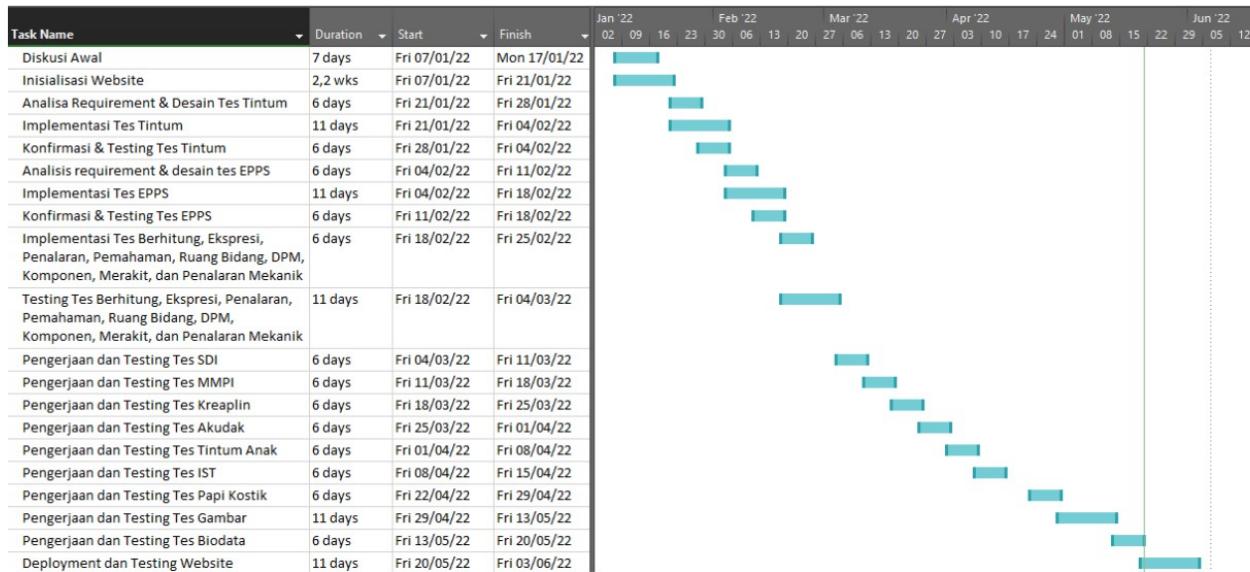
1.7 Penanggung Jawab

Dalam pengembangan website ini terdapat beberapa penanggung jawab untuk tiap bagian, sebagai berikut:

1. Project Manager : Lawrence Patrick
2. Desain User Interface : Widean Nagari
3. Code : Lawrence Patrick
4. Dokumentasi : Widean Nagari

1.8 Gantt Chart

Dalam pengembangan website tes psikologi ini, diperlukan penjadwalan pengerjaan tiap task yang ada agar dapat diselesaikan tepat waktu. Berikut jadwal pelaksanaan dalam satuan waktu minggu:



**Gambar 1. 2
Gantt Chart**

1.9 Sistematika Pembahasan

Untuk memperoleh pembahasan yang sistematis, penulis perlu menyusun sistematika sedemikian rupa sehingga dapat menunjukkan hasil penelitian yang baik dan mudah dipahami. Maka penulis akan mendeskripsikan sistematika penulisan sebagai berikut:

- BAB I : PENDAHULUAN

Pada bab ini, dibahas latar belakang dan alasan pembuatan website, tujuan pembuatan website, teori-teori yang menunjang pembuatan website, arsitektur program, dan penjelasan cara berjalannya proyek ini dimulai dari metodologi yang dipakai, penanggung jawab tiap bagian, sampai gantt chart yang menjadi pedoman/acuan tugas yang akan dikerjakan oleh tiap anggota tim.

- BAB II : ANALISA SISTEM

Pada bab ini, dibahas tentang fact finding, fitur yang dibutuhkan oleh pengguna website tes psikologi, dan keseluruhan sistem yang akan dibuat dalam proyek ini.

- BAB III : SISTEM DESAIN

Pada bab ini, akan dibahas tentang desain sistem pada website tes psikologi PT. Adiputro Wirasejati. yang meliputi desain arsitektur, desain interface, serta desain database.

- BAB IV : IMPLEMENTASI

Setelah spesifikasi dan sistem telah ditentukan, maka tahap selanjutnya adalah mengimplementasikan ke dalam program. Pada bab ini akan dijelaskan mengenai implementasi tentang setiap fitur yang dimiliki oleh website tes psikologi PT. Adiputro Wirasejati.

- BAB V : UJI COBA

Setelah dilakukan tahapan implementasi, langkah selanjutnya adalah melakukan pengujian pada sistem. Pada bab ini akan dijelaskan tentang analisis hasil uji coba website tes psikologi PT. Adiputro Wirasejati.

- BAB VI : PENUTUP

Pada bab ini akan dibahas tentang kesimpulan dan saran dari pengembangan website tes psikologi PT. Adiputro Wirasejati. yang nantinya dapat dikembangkan lebih lanjut dalam perancangan berikutnya.

- BAB VII : USER MANUAL

Bab ini berisi user manual atau panduan pengguna. Pada bab ini akan dibahas mengenai cara menggunakan program bagi pengguna. Adanya user manual diharapkan dapat mempermudah pengguna agar tidak bingung saat menggunakan program ini.

BAB II

ANALISA SISTEM

Pada bab ini, dibahas tentang fact finding, fitur yang dibutuhkan oleh pengguna website tes psikologi, dan keseluruhan sistem yang akan dibuat dalam proyek ini.

2.1. Profil Perusahaan

PT Adiputro Wirasejati atau lebih dikenal sebagai Adi Putro adalah sebuah perusahaan karoseri bus asal Malang, Jawa Timur. Adi Putro didirikan oleh tiga bersaudara yang bernama Andreas Jethrokusumo, Jesse Jethrokusumo dan David Jethrokusumo pada tahun 1975 di Kota Malang, Jawa Timur yang hingga kini masih dijadikan kantor pusatnya. Adi Putro telah membuat banyak bodi bus yang digunakan oleh banyak perusahaan-perusahaan otobus ternama di Indonesia seperti Lorena, Pahala Kencana, ALS, dan lainnya.

PT Adiputro WIRASEJATI beroperasi di Malang, tepatnya di 35 Jalan Raya Balearjosari, Kota Malang, Provinsi Jawa Timur. Karoseri buatan Adiputro memang terkenal luas dikarenakan produknya yang inovatif, kontrol kualitas tinggi, dan teknologi mutakhir dengan beberapa produk unggulan terbaru seperti Jetbus MCT, Jetbus 2 MC, Jetbus 2+ MD, Jetbus 2+ HD, dan lain-lainnya.

Demi menjaga kualitasnya, tim R&D dan insinyur dari karoseri Adiputro pun secara berkala belajar di Jerman untuk memperbarui diri dengan teknologi baru. Adiputro terus meningkatkan keterampilan baru dalam pekerjaan mereka serta menambah mesin-mesin baru. Pada tahun 1982, Adiputro berinvestasi dengan membeli mesin-mesin baru untuk menciptakan kualitas FULL PRESSED BODY dalam karoseri. Bahkan hingga sekarang, karoseri Adiputro siap menyambut tren baru dalam dunia otomotif dimana untuk pertama kalinya dalam sejarah, pasar mobil Indonesia pada tahun 2016 mulai menawarkan bus dengan kategori berat di atas 24 ton. Adaptasi terhadap bisnis baru tersebut sudah mulai dilakukan agar karoseri Adiputro bisa ikut berkembang seiring berkembangnya zaman

2.2. Fact Finding

Pada pengembangan website ini digunakan teknik fact finding wawancara. Beberapa pertanyaan diajukan kepada tim Adiputro guna mengetahui spesifikasi yang dibutuhkan dalam pengerjaan website psikotes ini. Berikut adalah beberapa pertanyaan yang diajukan.

1. Apakah tes yang nantinya akan dibuat bersifat online atau bisa dikerjakan secara offline juga?
2. Untuk sistem tesnya seperti apa?
3. Apakah peraturan atau sistem penilaian diberikan sebagai acuan penilaian?
4. Apakah kami harus membuat settingan hosting sendiri untuk website nya atau nanti hosting akan diurus oleh tim IT Adiputro?
5. Sistem pengerjaan tes per orang seperti apa?
6. Apakah satu orang dapat melakukan lebih dari satu tes?
7. Seperti apa sistem keamanan autentikasi setiap user yang ingin melakukan tes?
8. Tampilan dari website apakah disediakan atau harus mendesain sendiri?
9. Harus mendesain database seperti apa?
10. Soal-soal tes apakah nantinya diberikan atau kami harus membuat data dummy?

Dari hasil wawancara yang telah dilakukan, ditemukan bahwa cara kerja sistem saat ini adalah sebagai berikut.

Tes masuk psikotes yang dimiliki Adiputro sekarang masih dilakukan secara offline atau manual menggunakan kertas. Terdapat beberapa tes yang harus dikerjakan oleh pelamar kerja dan setiap tes memiliki beberapa bagian. Di setiap bagian, terdapat kumpulan pertanyaan yang nantinya harus dikerjakan. Pelamar kerja diberikan lembar soal beserta dengan lembar jawaban yang harus diisi. Pelamar diberikan waktu tertentu untuk mengerjakan tesnya menggunakan pulpen atau pensil. Beberapa contoh hasil tes yang telah dilakukan adalah sebagai berikut:

1. Tes Adkudag

Pada lembar jawaban tes adkudag persoalan 1, peserta dapat mengisi data diri pada bagian form data diri yang dapat dilihat pada gambar 2.1 nomor 1. Untuk menjawab pertanyaan, telah disediakan kolom-kolom yang dapat diisi dengan jawaban peserta (dapat dilihat pada gambar 2.1 nomor 2). Jawaban untuk persoalan ini adalah X atau

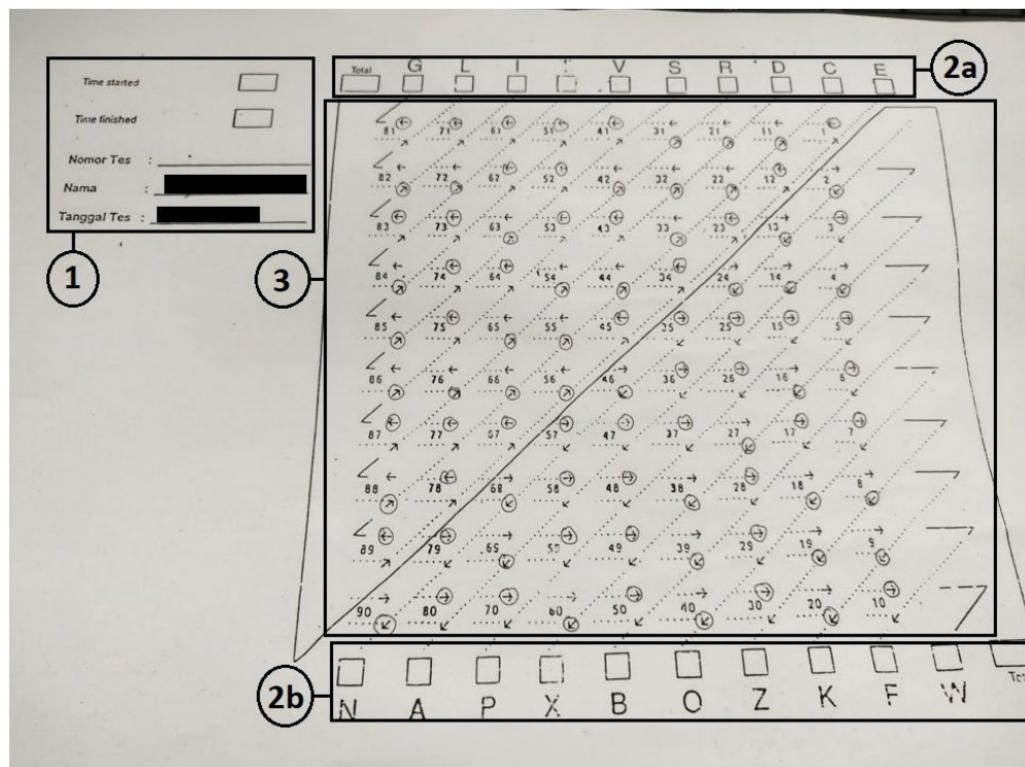
O. Pada gambar 2.1 nomor 3 dapat dilihat faktor dan hasil penilaian yang dihitung dan diisi secara manual.

LEMBAR JAWABAN	
Nomer : Umur : tahun	NAMA : Tgl : Pendidikan : Tester :
PERCOBAAN ADKUDAG I	
1. <input checked="" type="radio"/> 26. <input checked="" type="radio"/> 54. <input checked="" type="radio"/> 76. 101. 126. 2. <input checked="" type="radio"/> 27. <input checked="" type="radio"/> 52. <input checked="" type="radio"/> 77. 102. 127. 3. <input checked="" type="radio"/> 28. <input checked="" type="radio"/> 53. <input checked="" type="radio"/> 78. 103. 128. 4. <input checked="" type="radio"/> 29. <input checked="" type="radio"/> 54. <input checked="" type="radio"/> 79. 104. 129. 5. <input checked="" type="radio"/> 30. <input checked="" type="radio"/> 55. <input checked="" type="radio"/> 80. 105. 130. 6. <input checked="" type="radio"/> 31. <input checked="" type="radio"/> 56. <input checked="" type="radio"/> 81. 106. 131. 7. <input checked="" type="radio"/> 32. <input checked="" type="radio"/> 57. <input checked="" type="radio"/> 82. 107. 132. 8. <input checked="" type="radio"/> 33. <input checked="" type="radio"/> 58. <input checked="" type="radio"/> 83. 108. 133. 9. <input checked="" type="radio"/> 34. <input checked="" type="radio"/> 59. <input checked="" type="radio"/> 84. 109. 134. 10. <input checked="" type="radio"/> 35. <input checked="" type="radio"/> 60. <input checked="" type="radio"/> 85. 110. 135. 11. <input checked="" type="radio"/> 36. <input checked="" type="radio"/> 61. <input checked="" type="radio"/> 86. 111. 136. 12. <input checked="" type="radio"/> 37. <input checked="" type="radio"/> 62. <input checked="" type="radio"/> 87. 112. 137. 13. <input checked="" type="radio"/> 38. <input checked="" type="radio"/> 63. <input checked="" type="radio"/> 88. 113. 138. 14. <input checked="" type="radio"/> 39. <input checked="" type="radio"/> 64. <input checked="" type="radio"/> 89. 114. 139. 15. <input checked="" type="radio"/> 40. <input checked="" type="radio"/> 65. <input checked="" type="radio"/> 90. 115. 140. 16. <input checked="" type="radio"/> 41. <input checked="" type="radio"/> 66. <input checked="" type="radio"/> 91. 116. 141. 17. <input checked="" type="radio"/> 42. <input checked="" type="radio"/> 67. <input checked="" type="radio"/> 92. 117. 142. 18. <input checked="" type="radio"/> 43. <input checked="" type="radio"/> 68. <input checked="" type="radio"/> 93. 118. 143. 19. <input checked="" type="radio"/> 44. <input checked="" type="radio"/> 69. <input checked="" type="radio"/> 94. 119. 144. 20. <input checked="" type="radio"/> 45. <input checked="" type="radio"/> 70. <input checked="" type="radio"/> 95. 120. 145. 21. <input checked="" type="radio"/> 46. <input checked="" type="radio"/> 71. <input checked="" type="radio"/> 96. 121. 146. 22. <input checked="" type="radio"/> 47. <input checked="" type="radio"/> 72. <input checked="" type="radio"/> 97. 122. 147. 23. <input checked="" type="radio"/> 48. <input checked="" type="radio"/> 73. 98. 123. 148. 24. <input checked="" type="radio"/> 49. <input checked="" type="radio"/> 74. 99. 124. 149. 25. <input checked="" type="radio"/> 50. <input checked="" type="radio"/> 75. 100. 125. 150.	
R.S. = 70	W.S. = 16
32/5/6,4 1/c	

Gambar 2.1
Lembar Jawaban Tes Adkudag Persoalan 1

2. Tes Papi Kostick

Pada lembar jawaban tes papi kostick, peserta dapat mengisi data diri pada bagian form data diri yang dapat dilihat pada gambar 2.2 nomor 1. Untuk menjawab pertanyaan, telah disediakan kolom-kolom yang dapat diisi dengan jawaban peserta (dapat dilihat pada gambar 2.2 nomor 3). Untuk menjawab tes ini, peserta dapat melengkapi panah horizontal maupun vertikal. Jawaban diisi secara menurun dan dari kanan ke kiri. Pada gambar 2.2 nomor 2a dan 2b dapat dilihat kolom penilaian yang nantinya akan diisi secara manual.



Gambar 2.2
Lembar Jawaban Tes Papi Kostick

3. Tes Ekspresi

Pada lembar jawaban tes ekspresi, peserta dapat mengisi data diri pada bagian form data diri yang dapat dilihat pada gambar 2.3 nomor 1. Pada bagian bawah form data diri terdapat petunjuk untuk mengerjakan tes ini. Untuk menjawab pertanyaan, telah disediakan nomor soal beserta pilihan a, b, dan c (dapat dilihat pada gambar 2.3 nomor 2). Pada tes ini, peserta akan diberikan 3 jenis pernyataan, peserta dapat memberikan tanda X untuk pernyataan paling jelek dan tanda O untuk pernyataan paling baik. Pada gambar 2.3 nomor 3 dapat dilihat hasil penilaian dan norma yang dihitung dan diisi secara manual.

LEMBAR JAWABAN EKSPRESI

NAMA : ██████████
UMUR : ██████████
PENDIDIKAN : ██████████
TANGGAL TEST : ██████████
PETUNJUK : ██████████

Beri tanda lingkaran (O) untuk pilihan yang paling baik
Beri tanda lingkaran (X) untuk pilihan yang paling jelek

1. X b (c)
2. a (b) X
3. (d) X c
4. a (b) X
5. (a) b X
6. a (b) (c)
7. a (b) (c)
8. a (b) X
9. (a) X c
10. X (b) c
11. (a) b X
12. X b (c)
13. a (b) X
14. X b (c)
15. (a) X c
16. a (b) X
17. X (b) c
18. a (b) X
19. (a) X c

13/C

Gambar 2.3
Lembar Jawaban Tes Ekspresi

Terdapat berbagai macam jenis tes seperti tes pilihan ganda, tes mencocokkan gambar, tes menggambar, tes menghafal, dan lain-lain. Setiap tes memiliki keunikan masing-masing dan memiliki indeks pengukuran psikis si pekerja tes. Dalam setiap tes, terdapat norma atau aturan yang perlu diikuti untuk mengukur berbagai macam kondisi psikis. Norma tersebut tertulis di sebuah buku kumpulan norma sebagai pedoman.

Setelah pelamar selesai mengerjakan tes, tim Adiputro harus melakukan pengecekan soal secara manual untuk setiap soal lalu menghitung norma dari hasil jawaban pelamar. Diakui pekerjaan tersebut memakan waktu dan sangat membosankan. Maka dari itu diperlukannya sistem baru yang lebih otomatis agar lebih memudahkan pelaksanaan tes masuk adiputro.

2.3. Deskripsi Sistem Baru

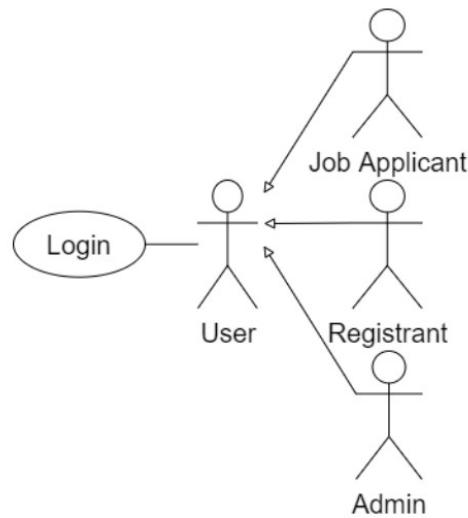
Sistem baru yang akan dibuat bertujuan untuk membuat cara kerja sistem yang lama menjadi lebih efisien dan efektif. Berikut adalah cara kerja sistem yang baru:

1. Use Case

Use case adalah teknik penemuan kebutuhan perangkat lunak yang dikenalkan pertama kali dalam metode pendekatan berbasis objek yang dikembangkan oleh Jacobson pada tahun 1990an. Use case mendeskripsikan tentang tingkah laku sistem di berbagai kondisi dan bagaimana sistem tersebut menanggapi permintaan pengguna. Use case didefinisikan dari sudut pandang aktor yang terlibat. Aktor merupakan orang atau alat yang menggunakan sistem tersebut atau bisa juga sebagai apapun yang berinteraksi dengan sistem. Dari hasil analisis website yang ada maka use case diagram untuk website tes psikologi PT Adiputro Wirasejati dapat dilihat pada poin dibawah:

a. User

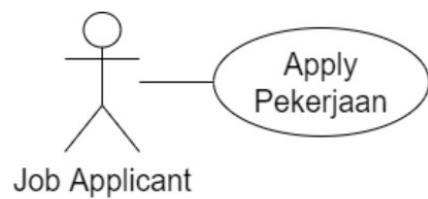
Secara general, semua user dapat melakukan login. Login bertujuan agar user dapat masuk ke halaman dashboard yang berisi daftar tes yang harus dikerjakan dengan email dan login token yang sudah didaftarkan admin. Use case dari user secara general dapat dilihat pada gambar 2.4.



**Gambar 2. 4
Use Case User**

b. Job Applicant

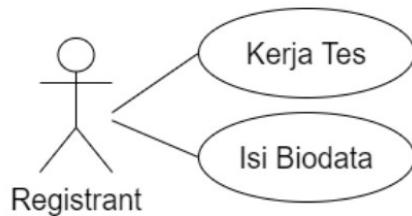
Job applicant dapat melakukan apply pada lowongan pekerjaan yang sedang dibuka. Use case dari Job applicant dapat dilihat pada gambar 2.5.



Gambar 2. 5
Use Case Job Applicant

c. User Biasa

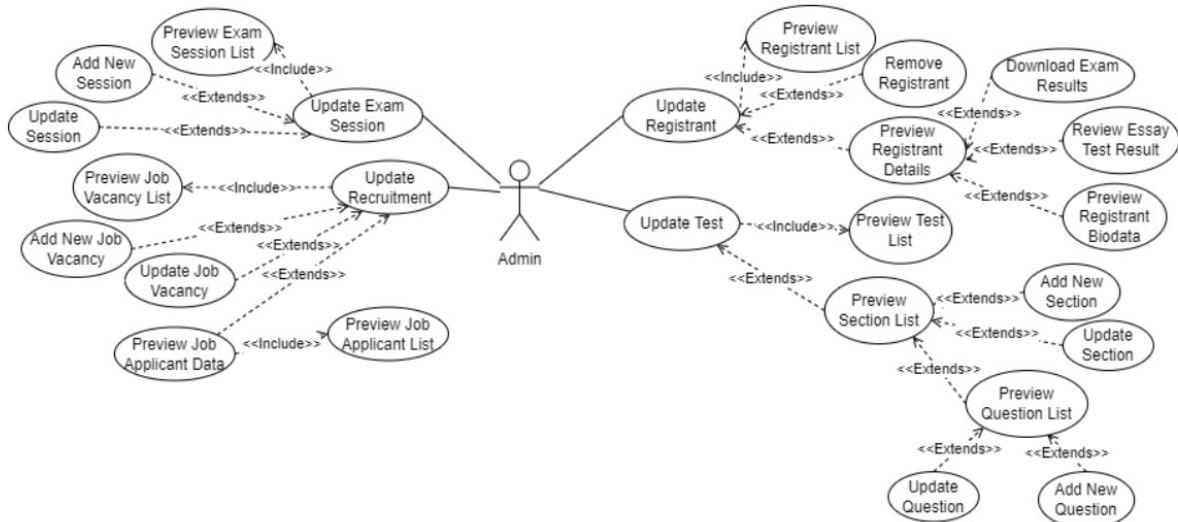
User/peserta tes yang sudah melakukan login dapat melakukan beberapa hal, antara lain memilih tes yang ingin dikerjakan, mengerjakan tes, dan Mengisi Biodata. Use case dari user/peserta tes dapat dilihat pada gambar 2.6.



Gambar 2. 6
Use Case User Biasa

d. Admin

Admin dapat melakukan beberapa hal, antara lain mengupdate user, mengupdate test, mengupdate question, mengupdate session, mengupdate exam session, mengupdate exam result, dan mengupdate data recruitment. Use case dari Admin forum dapat dilihat pada gambar 2.7.



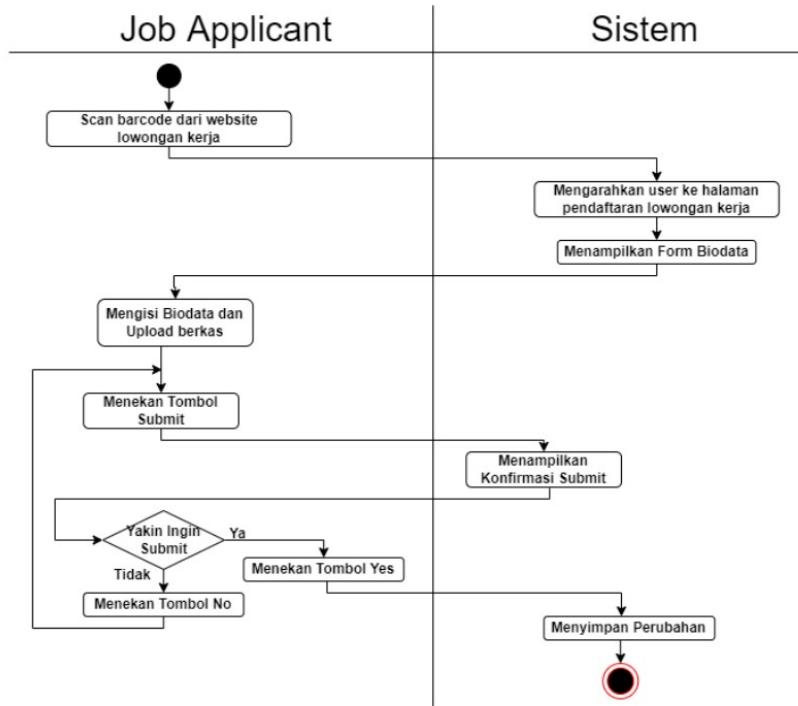
**Gambar 2. 7
Use Case Admin**

2. Activity Diagram

Activity Diagram adalah bentuk visual dari alir kerja yang berisi aktivitas dan tindakan, yang juga dapat berisi pilihan, pengulangan, dan *concurrency*. Dalam Unified Modelling Language (UML), diagram aktivitas dibuat untuk menjelaskan aktivitas komputer maupun alur aktivitas dalam organisasi secara garis besar. Berdasarkan use case yang sudah dijelaskan pada poin sebelumnya, berikut activity diagram dari website tes psikologi PT. Adiputro:

a. Job Applicant - Apply Pekerjaan

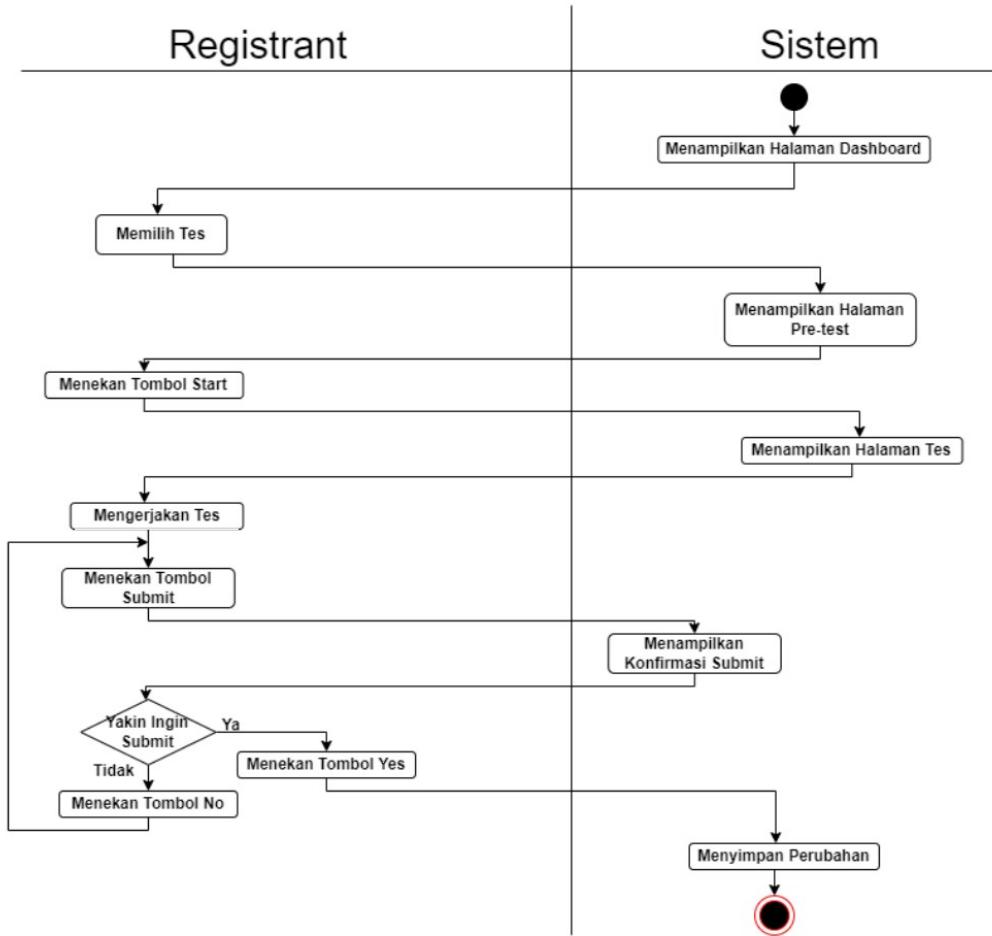
Untuk melamar pekerjaan, mula-mula pelamar harus scan barcode yang ada pada lowongan kerja yang diminati, lalu sistem akan mengarahkan user ke halaman pendaftaran lowongan kerja dan menampilkan form biodata. User dapat mengisi data diri beserta upload berkas-berkas yang dibutuhkan. Apabila user melakukan submit pada form tersebut, sistem akan melakukan cek apakah email yang digunakan sudah pernah dipakai, karena email yang dipakai harus unik. Apabila email sudah ada sebelumnya dan status nya masih aktif, maka user harus menginputkan email lainnya, apabila email belum ada maka sistem akan menambahkan data pendaftar baru. Gambar activity diagram untuk apply pekerjaan dapat dilihat pada gambar 2.8



Gambar 2. 8
Activity Diagram Job Applicant – Apply Pekerjaan

b. Registrant - Kerja Tes

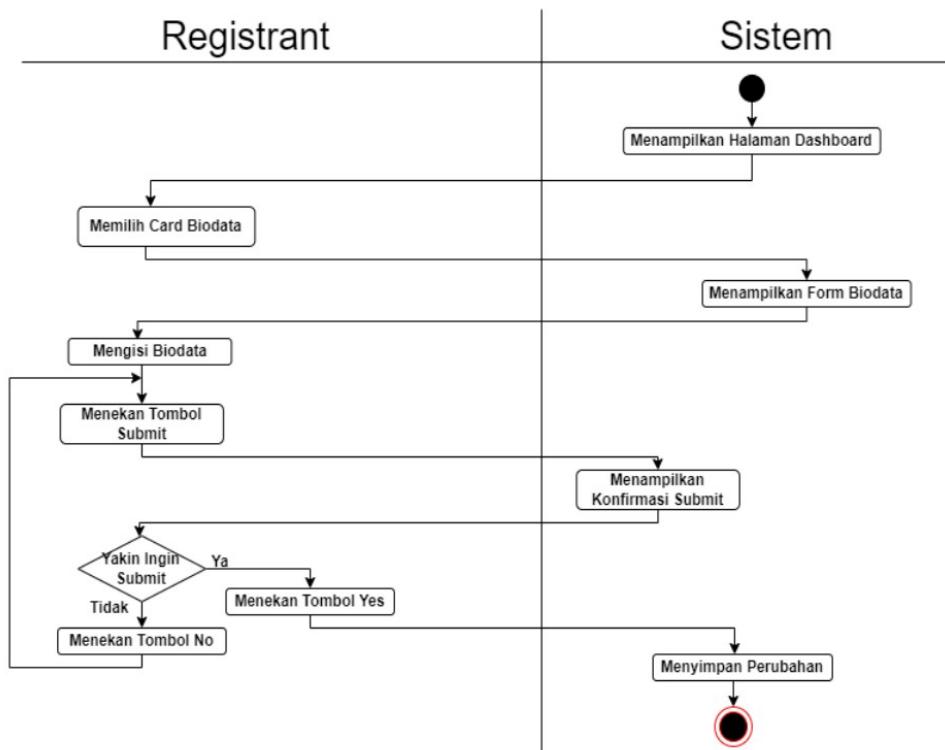
Untuk mengerjakan tes, mula-mula registrant harus melakukan login dengan menggunakan email saat pendaftaran dan test token yang telah diberikan , lalu sistem akan mengarahkan user ke halaman dashboard. User dapat memilih tes mana yang akan dikerjakan. Bila user menekan tes yang akan dikerjakan, maka sistem akan mengarahkan user ke halaman pre-Test dimana pada halaman ini akan ditampilkan penjelasan cara mengerjakan tes. Apabila user sudah siap maka user dapat menekan tombol start dan sistem akan menampilkan halaman tes. Berikutnya user dapat mengerjakan semua soal yang sudah ditampilkan. Apabila user melakukan submit pada tes tersebut atau waktunya sudah habis, maka sistem akan menampilkan pesan konfirmasi, bila user menekan tombol yakin untuk submit maka sistem akan menyimpan data jawaban user serta melakukan perhitungan nilai. Gambar activity diagram untuk kerja tes dapat dilihat pada gambar 2.9



Gambar 2. 9
Activity Diagram Registrant – Kerja Tes

c. Registrant - Isi Biodata

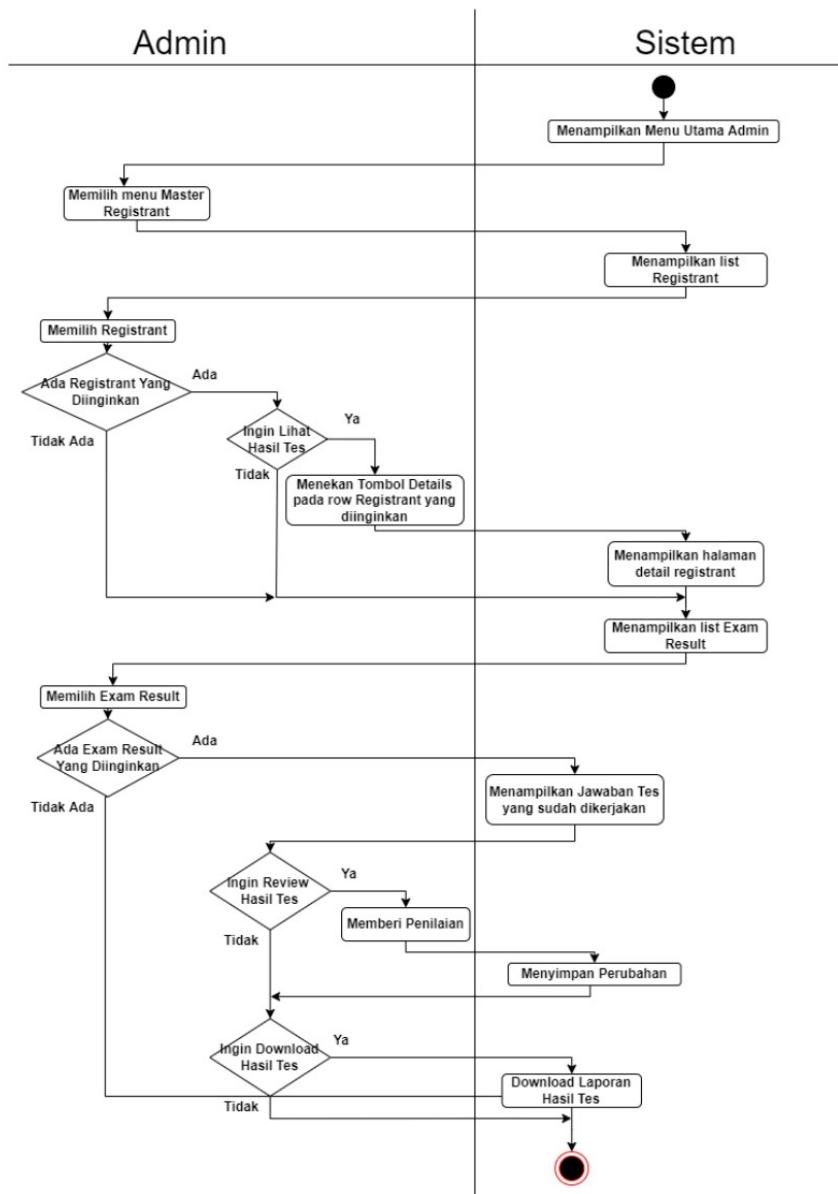
Untuk mengisi biodata, mula-mula registrant harus melakukan login dengan menggunakan email saat pendaftaran dan test token yang telah diberikan , lalu sistem akan mengarahkan user ke halaman dashboard. User dapat memilih card biodata dan sistem akan mengarahkan user ke halaman pengisian biodata. Berikutnya user dapat mengisi semua data yang diminta pada form. Apabila user melakukan submit pada form tersebut maka sistem akan menampilkan pesan konfirmasi, bila user menekan tombol yakin untuk submit maka sistem akan menyimpan data biodata user. Gambar activity diagram untuk isi biodata dapat dilihat pada gambar 2.10



Gambar 2. 10
Activity Diagram Registerant – Isi Biodata

d. Admin - Master Registrant

Untuk mengakses master registrant, mula-mula admin harus login ke dalam website sebagai admin, lalu sistem akan menampilkan halaman dashboard admin. Admin dapat memilih tab master registrant dan sistem akan menampilkan list semua registrant yang sudah terdaftar. Admin dapat memilih user mana yang ingin dilihat detailnya. Lalu sistem akan menampilkan halaman detail user beserta daftar tes yang sudah dikerjakan. Admin dapat memberikan penilaian terhadap tes yang berjenis essay. Admin juga dapat download laporan hasil tes. Gambar activity diagram untuk master registrant dapat dilihat pada gambar 2.11.

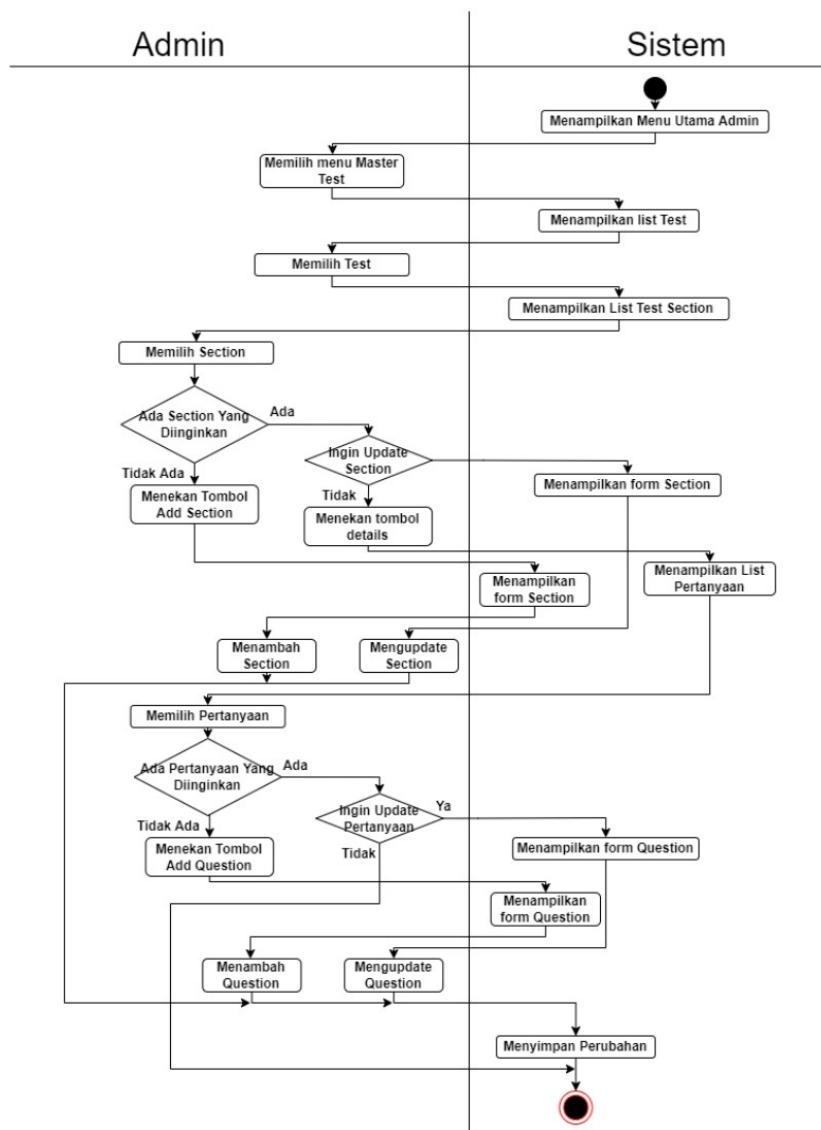


Gambar 2. 11
Activity Diagram Admin – Master Registrant

e. Admin - Master Test

Untuk mengakses master test, mula-mula admin harus login ke dalam website sebagai admin, lalu sistem akan menampilkan halaman dashboard admin. Admin dapat memilih tab master test dan sistem akan menampilkan list semua test yang sudah ditambahkan dalam bentuk drop down. Admin dapat memilih tes dan sistem akan menampilkan semua section dari test tersebut. Admin dapat menambahkan dan mengupdate section dengan menekan tombol add section

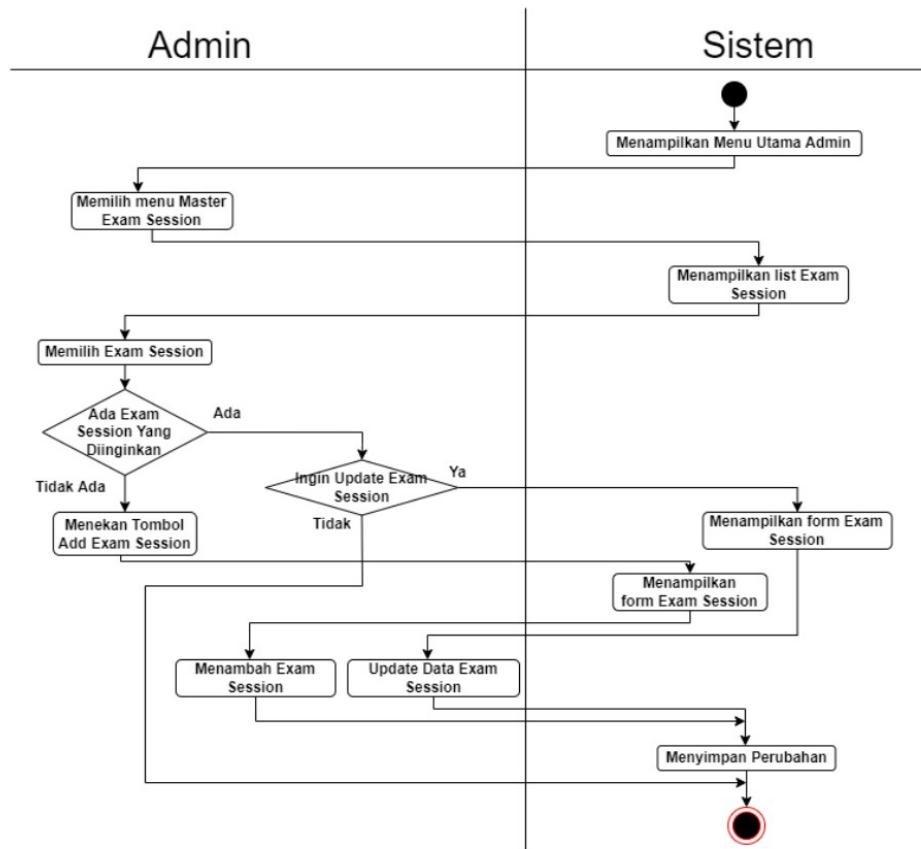
untuk menambah section dan update section untuk mengupdate section, lalu sistem akan menampilkan form section. Admin juga dapat melihat semua pertanyaan yang ada pada section tersebut dengan menekan tombol details pada baris section. Admin dapat menambahkan dan mengupdate pertanyaan yang ada dengan menekan tombol add question untuk menambah pertanyaan dan update question untuk mengupdate question, lalu sistem akan menampilkan form question. Gambar activity diagram untuk master test dapat dilihat pada gambar 2.12.



Gambar 2. 12
Activity Diagram Admin – Master Test

f. Admin - Update Session

Untuk mengupdate session, mula-mula admin harus login ke dalam website sebagai admin, lalu sistem akan menampilkan halaman dashboard admin. Admin dapat memilih tab master session dan sistem akan menampilkan list semua session yang sudah ditambahkan. Admin dapat menambahkan session baru dengan menekan tombol Buat Session Baru, sistem akan menampilkan modal buat session dan admin dapat menambahkan session baru. Admin juga dapat memilih session yang ingin di update dengan menekan tombol update pada row session, lalu sistem akan menampilkan modal update session beserta data saat ini, lalu admin dapat mengganti status session menjadi non-aktif. Gambar activity diagram untuk update session dapat dilihat pada gambar 2.13.

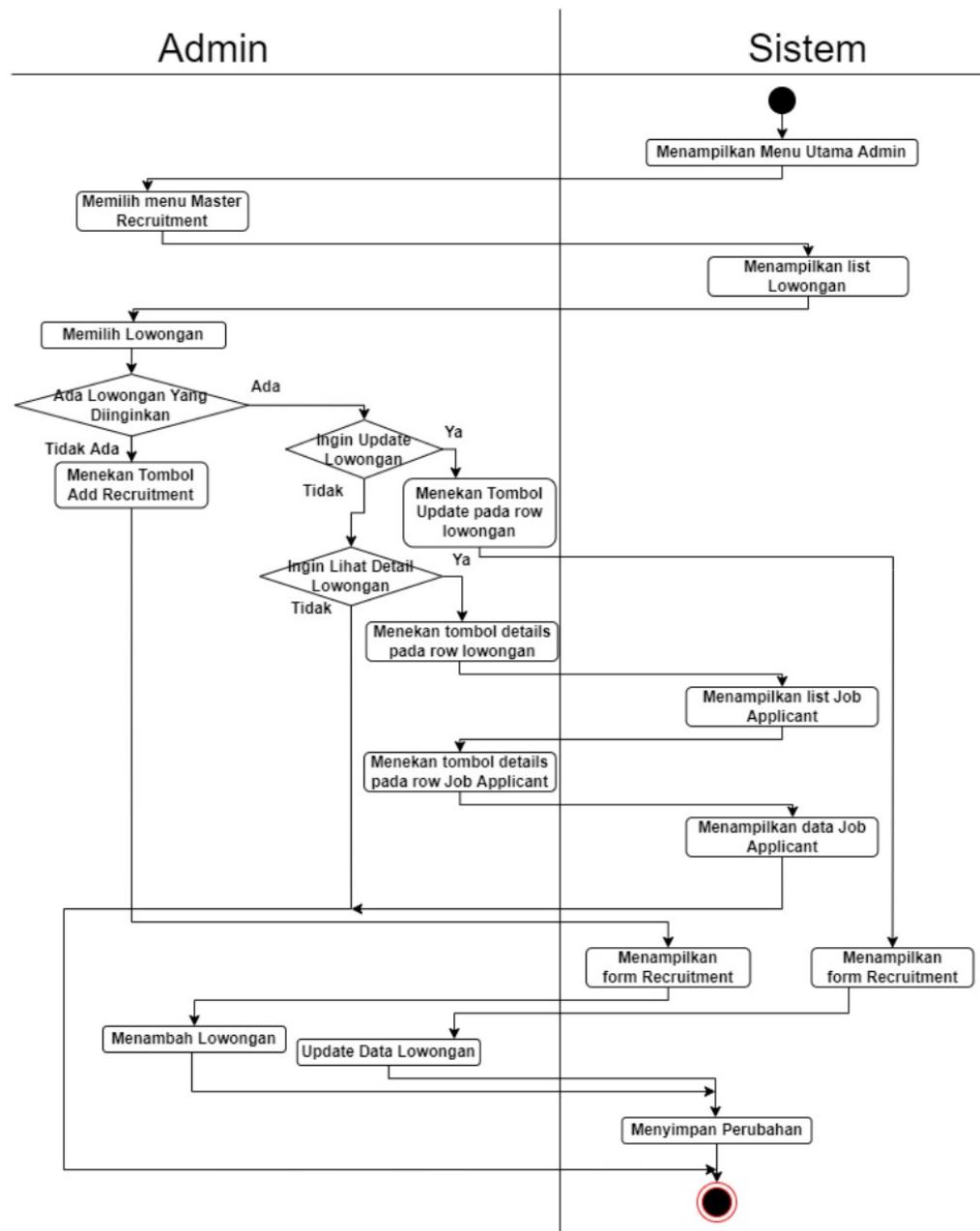


Gambar 2. 13
Activity Diagram Admin – Update Session

g. Admin - Master Recruitment

Untuk mengakses Master Recruitment, mula-mula admin harus login ke dalam website sebagai admin, lalu sistem akan menampilkan halaman dashboard admin. Admin dapat memilih tab master recruitment dan sistem akan menampilkan list semua lowongan kerja yang sudah dibuat. Admin dapat menambahkan lowongan baru dengan menekan tombol Buat Lowongan Baru, sistem akan menampilkan modal buat lowongan dan admin dapat menambahkan lowongan baru. Admin juga dapat menutup lowongan dengan menekan tombol update pada baris lowongan yang diinginkan, lalu sistem akan menampilkan modal update lowongan beserta data saat ini, lalu admin dapat mengganti status lowongan menjadi non-aktif.

Admin juga dapat melihat daftar pelamar kerja dengan menekan tombol detail pada lowongan yang diinginkan dan sistem akan menampilkan daftar pelamar kerja. Admin dapat melihat data diri dan berkas pelamar kerja dengan menekan tombol detail pada baris data pelamar. Gambar activity diagram untuk master recruitment dapat dilihat pada gambar 2.14.



Gambar 2. 14
Activity Diagram Admin – Master Recruitment

2.4. Kebutuhan Software

Setelah menganalisa permintaan client untuk pembuatan website, maka ditemukan beberapa kebutuhan sistem yang baru untuk mengembangkan website tes psikologi PT. Adiputro Wirasejat, antara lain:

1. Sistem login hanya sekali

Fitur ini bertujuan agar setiap session user hanya dapat melakukan login sebanyak 1 kali pada satu waktu yang sama, sehingga dapat mengurangi peluang user berbuat curang dengan mengerjakan di banyak device dengan orang yang berbeda.

2. Tes psikologi

Fitur ini adalah fitur inti dari website ini. Terdapat 20 macam alat tes yang dapat diassign kan kepada para calon pegawai. Nantinya user dapat mengerjakan tes yang telah diberikan. Pada tes ini akan diberikan waktu ±1 hari, sehingga pada waktu tersebut user bebas ingin mengerjakan kapanpun, namun saat waktu sudah habis maka user tidak lagi dapat mengerjakan tes dan otomatis terlogout.

3. Biodata

Fitur ini bertujuan agar calon pegawai dapat mengisikan data diri yang dibutuhkan perusahaan. Untuk mengakses form biodata, user dapat menekan card biodata, dimana card biodata ini nantinya akan dibuat sama seperti card test yang lain.

4. Tampilan Responsive

Fitur ini memungkinkan peserta tes mengerjakan tes yang diberikan menggunakan pc/laptop maupun menggunakan smartphone/tablet.

5. Laporan hasil tes

Fitur ini memungkinkan admin untuk melihat laporan dari semua tes yang sudah dikerjakan oleh setiap calon pegawai. Laporan digenerate secara otomatis berdasarkan data hasil tes yang dihitung secara otomatis pula.

6. Download laporan

Fitur ini memungkinkan admin untuk mengunduh hasil laporan dari semua tes yang sudah dikerjakan oleh setiap calon pegawai. Laporan nantinya akan dibuat dalam file berekstensi .pdf.

7. Assign tes

Fitur ini memungkinkan admin untuk memilih tes mana saja yang akan diberikan pada calon pegawai. Terdapat 3 set tes tetap (untuk SMK, S1, dan S1-Teknik) dan 1 set tes opsional. Set tes tetap ini merupakan set yang berisi tes yang sudah fix dan dapat dipilih sesuai dengan pendidikan terakhir calon pegawai. Sedangkan set tes opsional akan berisi beberapa tes yang dapat dipilih bebas.

8. Recruitment

Fitur ini memungkinkan admin untuk membuka lowongan kerja baru. Setiap lowongan kerja akan memiliki QR Code masing-masing yang mana dapat diunduh oleh admin untuk diposting. Fitur QR Code ini bertujuan agar setiap pelamar yang scan QR tersebut dapat diarahkan ke halaman pengisian data diri dan berkas-berkas yang dibutuhkan untuk melamar pekerjaan. Admin dapat melihat semua data pelamar yang telah melamar pada lowongan tersebut. Admin juga dapat menutup lowongan kerja yang ada apabila sudah ada pelamar yang diterima.

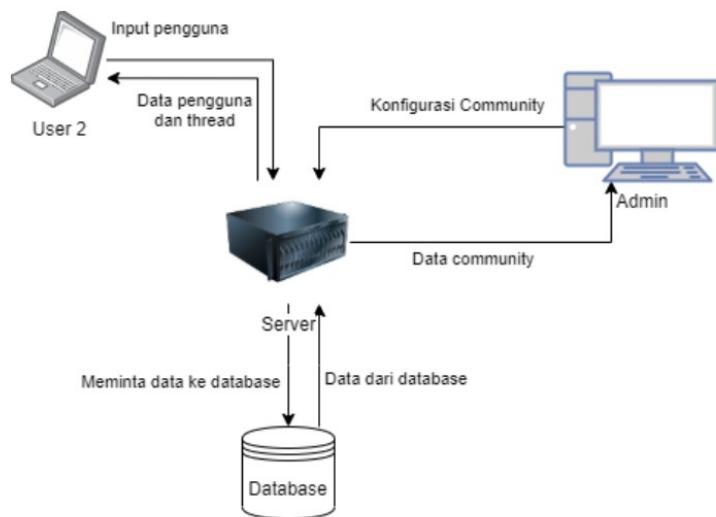
BAB III

DESAIN SISTEM

Pada bab ini, akan dibahas tentang desain sistem pada website tes psikologi PT. Adiputro Wirasejati. yang meliputi desain arsitektur, desain interface, serta desain database.

3.1 Desain Arsitektur

Dalam perancangan arsitektur sistem website tes psikologi PT. Adiputro Wirasejati, terdapat beberapa komponen yang berperan, yaitu device user yang berperan sebagai sarana bagi user untuk mengerjakan tes, device admin yang berperan sebagai sarana bagi admin untuk mengatur data pada website dan mendapatkan report dari tiap hasil tes, server sebagai pusat kendali sistem, dan database sebagai pusat dari semua data pada website. Gambaran arsitektur sistem dari website PT. Adiputro Wirasejati adalah sebagai berikut:



Gambar 3. 1
Arsitektur Sistem

3.2 Desain Database

Perancangan/Desain Database adalah proses untuk menentukan isi dan pengaturan data yang dibutuhkan untuk mendukung berbagai rancangan sistem. Tujuan Perancangan

Database adalah untuk memenuhi informasi yang berisikan kebutuhan-kebutuhan user secara khusus dan aplikasi-aplikasinya dan juga memudahkan pengertian struktur informasi.

3.2.1 Desain Entity Relationship Diagram

Website Adiputro Psikotes menggunakan database dengan format SQL. Berikut akan dijelaskan mengenai Entity Relationship Diagram (ERD) yang menunjukkan relasi antar yang biasa disebut Conceptual Data Model (CDM).

ERD dibuat untuk menggambarkan atau membuat model suatu database dengan diagram yang sederhana, sehingga memudahkan dalam membuat sebuah database yang kompleks maupun yang sederhana. Dengan ERD ini juga akan memudahkan pengelola yang merancang database tersebut dalam membuat perubahan di dalam database jika sewaktu-waktu terjadi. Berikut adalah ERD dari website Adiputro Psikotes.

Gambar ERD (CDM) dari website Adiputro Psikotes dapat dilihat pada Lampiran 1. Contextual Data Model yang kami buat cukup besar dan setiap Entity yang ada akan berhubungan dengan Entity lainnya. Seorang pelamar pekerjaan nantinya akan dibuatkan sebuah Exam Session menggunakan email yang sudah mereka kumpulkan ke Adiputro. Setiap Exam Session memiliki unique email dan test token yang bertugas sebagai username dan password. Test Token akan di-generate kata random sepanjang 23.

Terdapat kumpulan Test yang mana setiap Test akan memiliki satu atau lebih Section. Relation dari Test dan Section adalah *one to many* (satu ke banyak). Pada setiap Section terdapat banyak Question yang nantinya pelamar akan mengerjakan soal-soal di sini. Relation dari Section dan Question sama seperti Tes dan Section yaitu *one to many* (satu ke banyak). Setiap Question pasti memiliki soal/petunjuk tetapi tidak semua soal memiliki pilihan jawaban. Hal ini dikarenakan ada beberapa Question yang bersifat isian ada yang bersifat multiple choice. Di setiap kolom question juga sudah ada kunci jawaban soal tersebut.

Setiap Exam Session akan memiliki waktu awal dan waktu akhir penggerjaan tes. Pelamar tidak bisa login ke website Adiputro Psikotes di luar tenggat waktu tes. Pada tenggat penggerjaan tes, setiap Exam Session akan diberikan kumpulan Test yang harus dikerjakan. Test apa saja yang harus dikerjakan setiap Exam Session disimpan pada tabel Exam_Session_Test. Exam_Session_Test memiliki hubungan one to many (satu ke banyak) ke Test dan Exam Session. Pelamar dapat mengerjakan Test yang mana saja terlebih dahulu

tetapi karena dalam satu Test terdapat banyak Section, pelamar harus urut mengerjakan Section-Section yang ada. Pada setiap Section sudah terdapat waktu penggerjaan yang bervariasi. Waktu penggerjaan tes ini tidak berjalan secara lokal tapi mengikuti jam server. Apabila waktu penggerjaan habis, maka pelamar akan ditendang keluar dari halaman penggerjaan tugas.

Hasil penggerjaan pelamar nantinya akan disimpan berdasarkan Question, Section, dan Test masing-masing yang sudah ia kerjakan. Setiap jawaban pertanyaan user akan disimpan di tabel Question_Result, section mana saja yang sudah dikerjakan pelamar serta berapa banyak pertanyaan yang benar dari section itu akan disimpan pada tabel Section_Result, dan setiap Test yang dikerjakan serta hasil perhitungan norma dan hasil keterangan psikotes Test tersebut akan disimpan pada tabel Test_Result. Hubungan relasi Question dan Question_Result, Section dan Section_Result, Test dan Test_Result adalah *one to many* (satu ke banyak). Selain itu, Test_Result dan Section_Result, Section_Result dan Question_Result juga memiliki hubungan *one to many* (satu ke banyak).

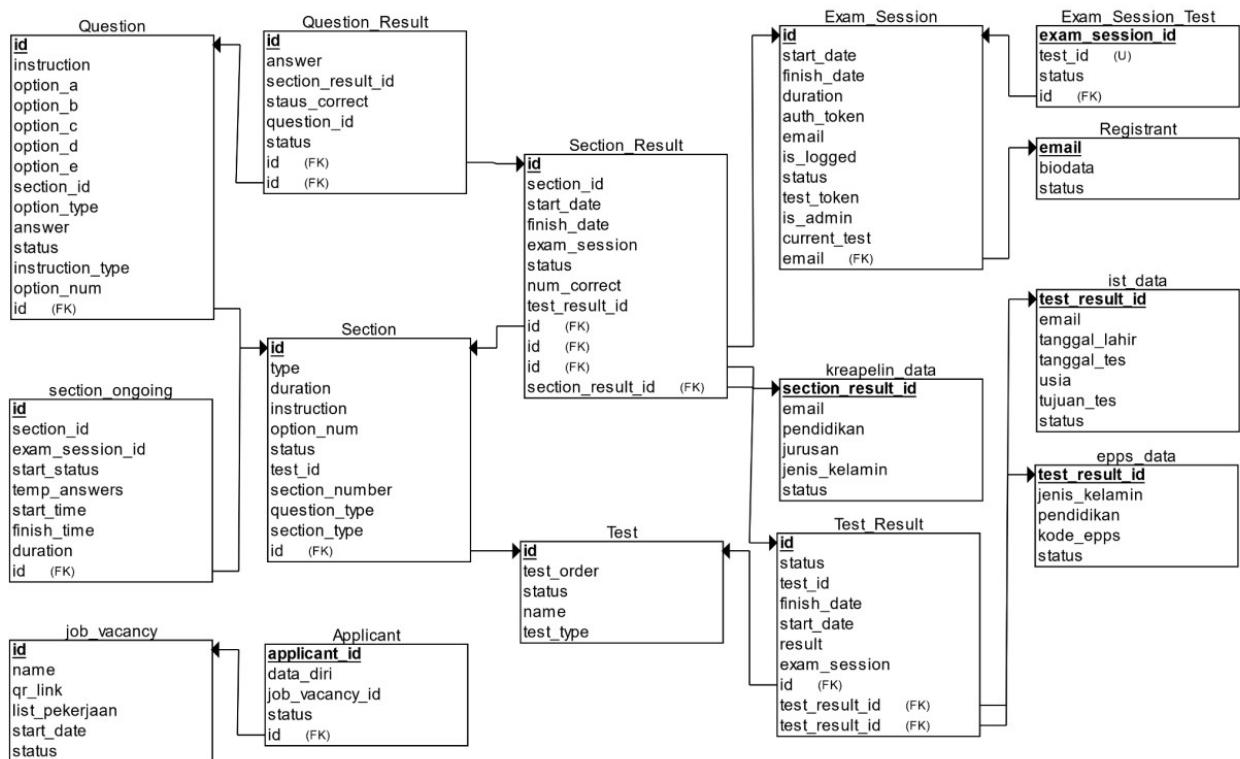
Khusus untuk beberapa Test, terdapat data tambahan yang perlu disimpan dikarenakan akan mempengaruhi norma dan bisa menjadi data tambahan sebagai pertimbangan tim HRD Adiputro. Test yang memerlukan data tambahan tersebut adalah Test IST dan Test Kreapelin. Kedua data test tersebut masing-masing akan masuk ke dalam IST_Data dan Kreapelin_Data. IST_Data akan memiliki hubungan relasi dengan Test_Result yaitu *one to many* (satu ke banyak). Sedangkan untuk Kreapelin_Data memiliki hubungan relasi dengan Section_Result karena setiap section dari Test Kreapelin memerlukan datanya masing-masing. Relasinya adalah *one to many* (satu ke banyak).

Pada database juga akan disimpan Section-Section mana saja yang sekarang sudah dikerjakan dan sudah dikerjakan oleh Exam Session. Maka dari itu dibuatlah tabel baru bernama Section_Ongoing. Selain itu, tabel ini juga dapat digunakan untuk perhitungan waktu tersisa dari penggerjaan section setiap Exam Session. Setiap Exam Session juga nantinya memiliki biodata secara general seperti umur, tanggal lahir, hobi, dan lain-lain. Maka dari itu setiap Exam Session akan memiliki sebuah relasi *many to one* (banyak ke satu) terhadap tabel Registrant.

Selain dari penggerjaan test, website ini juga dapat digunakan sebagai sarana pencarian pelamar kerja yang ingin mendaftarkan dirinya menjadi pekerja di Adiputro. Disediakan fitur

Job Vacancy yang digunakan sebagai lowongan pekerjaan yang buka dari dan dari hingga waktu tertentu. Siapapun dapat mendaftarkan dirinya dengan mengisi beberapa formulir yang sudah disediakan di halaman lamaran. Di halaman itu juga sudah tertulis posisi pekerjaan apa saja yang sedang dibuka pada lowongan tersebut. Setiap email hanya dapat mendaftarkan ke satu lowongan pekerjaan saja. Setiap email yang mendaftarkan diri akan disebut Applicant. Sehingga pada database, terdapat sebuah tabel bernama Applicant yang menyimpan semua data pelamar.

Setiap lowongan pekerjaan akan disimpan ke tabel Job_Vacancy. Job_Vacancy dan Applicant memiliki hubungan *one to many* (satu ke banyak). Pada tabel Job_Vacancy terdapat sebuah kolom bernama qr_link yang berisi link gambar QR Link yang ketika di-scan akan mengarahkan user ke halaman lamaran yang ada pada website Adiputro Psikotes. User tidak perlu login untuk mencoba melamar ke halaman ini.



**Gambar 3. 2
ERD Diagram (PDM)**

3.2.2 Desain Tabel

Pada sub-bab ini dijelaskan tentang properti atau atribut apa saja yang dimiliki oleh sebuah entitas atau tabel, dan penjelasan tentang sifat primary key, foreign key, dan sejenisnya pada setiap tabel. Dalam tabel ini pula terdapat field yang mencatat kapan baris dari tabel tersebut dibuat, diubah dan dihapus. Berikut akan dijelaskan mengenai tabel yang ada dalam website Adiputro Psikotes.

1. Tabel Job_Vacancy

Tabel Job_Vacancy merupakan sebuah tabel yang digunakan untuk menyimpan semua informasi lowongan pekerjaan yang sedang buka atau tidak buka. Tabel ini tidak memiliki foreign key ke tabel manapun.

**Tabel 3. 1
Tabel “Job Vacancy”**

Nama Field	Tipe	Costraint	Keterangan
id	BIGINT(10)	PRIMARY KEY	auto increment
name	VARCHAR(1000)	NOT NULL	
qr_link	VARCHAR(1000)	NOT NULL	- di-generate oleh program dan dimasukkan ke google drive. - yang disimpan adalah link google drive
list_pekerjaan	VARCHAR(1000)	NOT NULL	sekumpulan pekerjaan yang dipisahkan tanda ‘,’
start_date	DATETIME	NOT NULL	tanggal awal lowongan pekerjaan buka
status	INT(20)	NOT NULL	1: masih menerima pelamar 0: tidak menerima pelamar
createdAt	DATETIME	NOT NULL	tanggal data row pertama dibuat
updatedAt	DATETIME	NOT NULL	tanggal data row terakhir diperbarui

2. Tabel Applicant

Tabel Applicant merupakan sebuah tabel yang digunakan untuk menyimpan semua data milik pelamar pekerjaan yang mendaftarkan dirinya di lowongan pekerjaan

Job_Vacancy. Tabel ini memiliki foreign key job_vacancy_id dari tabel Job_Vacancy.

**Tabel 3.2
Tabel “Applicant”**

Nama Field	Tipe	Constraint	Keterangan
id	BIGINT(20)	PRIMARY KEY	auto increment
job_vacancy_id	BIGINT(10)	NOT NULL	- foreign Key Job_Vacancy(id). - Applicant melamar ke job vacancy yang mana
nama	VARCHAR(1000)	NOT NULL	
no_ktp	VARCHAR(1000)	NOT NULL	
tempat_lahir	VARCHAR(1000)	NOT NULL	
tanggal_lahir	DATE	NOT NULL	
jenis_kelamin	VARCHAR(1)	NOT NULL	L: laki-laki P: perempuan
alamat_domicili	VARCHAR(1000)	NOT NULL	
usia	INT(100)	NOT NULL	
status_perkawinan	VARCHAR(1000)	NOT NULL	
nomor_hp	VARCHAR(100)	NOT NULL	
email	VARCHAR(1000)	NOT NULL	
pendidikan_terakhir	VARCHAR(1000)	NOT NULL	
nama_sekolah	VARCHAR(1000)	NOT NULL	
jurusan	VARCHAR(1000)	NOT NULL	
posisi_dilamar	VARCHAR(1000)	NOT NULL	
status	INT(10)	NOT NULL	1: masih tersedia 0: tidak tersedia

lampiran_drive_id	VARCHAR(100)	NOT NULL	- link google drive tempat menyimpan portofolio atau data diri tambahan. - link otomatis dibuat website
portofolio	VARCHAR(255)	NOT NULL	- link portofolio dari pendaftar
createdAt	DATETIME	NOT NULL	tanggal data row pertama dibuat
updatedAt	DATETIME	NOT NULL	tanggal data row terakhir diperbarui

3. Tabel Exam_Session

Tabel Exam_Session merupakan sebuah tabel yang digunakan untuk menyimpan semua data sesi peserta tes setiap orang. Setiap Email dapat memiliki banyak sesi yang ditentukan waktu awal dan waktu akhir penggerjaannya. Tabel ini tidak memiliki foreign key ke tabel manapun.

Tabel 3. 3
Tabel “Exam Session”

Nama Field	Tipe	Costraint	Keterangan
id	BIGINT(20)	PRIMARY KEY	auto increment
email	VARCHAR(255)	NOT NULL	username peserta tes
start_date	DATETIME	NOT NULL	waktu awal penggerjaan tes
finish_date	DATETIME	NOT NULL	waktu akhir penggerjaan tes
duration	INT(10)	NOT NULL	selisih finish_date dan start_date dalam satuan menit
test_token	VARCHAR(500)	NOT NULL	- string random hasil generate program. - digunakan sebagai akses masuk seperti password di website
auth_token	VARCHAR(500)	NOT NULL	- string random hasil generate program. - digunakan sebagai token rahasia yang tidak perlu user ketahui untuk otorisasi pengguna web

is_logged	INT(10)	NOT NULL	1: user sedang logged in 0: user sedang logged out
is_admin	INT(10)	NOT NULL	1: user adalah admin 0: user biasa
current_test	BIGINT(10)	NOT NULL	- menandakan sedang mengerjakan tes apa. - default 0, selain itu berisi id Test
status	INT(10)	NOT NULL	1: masih tersedia 0: tidak tersedia
createdAt	DATETIME	NOT NULL	tanggal data row pertama dibuat
updatedAt	DATETIME	NOT NULL	tanggal data row terakhir diperbarui

4. Tabel Registrant

Tabel Registrant merupakan sebuah tabel yang digunakan untuk menyimpan semua data milik peserta tes psikotes dari setiap Exam_Session. Tabel ini memiliki foreign key email dari tabel Exam_Session.

**Tabel 3. 4
Tabel “Registrant”**

Nama Field	Tipe	Costraint	Keterangan
email	VARCHAR(255)	PRIMARY KEY	Foreign key ke Exam_Session field email
biodata	LONGTEXT	NOT NULL	Segala macam biodata dalam format json
status	INT(10)	NOT NULL	1: masih tersedia 0: tidak tersedia
createdAt	DATETIME	NOT NULL	tanggal data row pertama dibuat
updatedAt	DATETIME	NOT NULL	tanggal data row terakhir diperbarui

5. Tabel Test

Tabel Test merupakan sebuah tabel yang digunakan untuk menyimpan semua Test yang ada. Nantinya setiap Test akan memiliki banyak Section. Tabel ini tidak memiliki foreign key ke tabel lainnya.

**Tabel 3. 5
Tabel “Test”**

Nama Field	Tipe	Costraint	Keterangan
id	BIGINT(20)	PRIMARY KEY	auto increment
name	VARCHAR(255)	NOT NULL	Segala macam biodata dalam format json
test_order	VARCHAR(255)	NOT NULL	Urutan Test dikerjakan 1: in sequence 2 : not in sequence
test_type	VARCHAR(255)	NOT NULL	normal, EPPS, kreapelin, dll
status	INT(10)	NOT NULL	1: masih tersedia 0: tidak tersedia
createdAt	DATETIME	NOT NULL	tanggal data row pertama dibuat
updatedAt	DATETIME	NOT NULL	tanggal data row terakhir diperbarui

6. Tabel Exam_Session_Test

Tabel Test merupakan sebuah tabel yang digunakan untuk menandakan tes mana saja yang harus dikerjakan oleh setiap Exam_Session. Tabel ini memiliki foreign key ke tabel Test dan Exam_Session.

**Tabel 3. 6
Tabel “Exam_Session_Test”**

Nama Field	Tipe	Costraint	Keterangan
exam_session_id	VARCHAR(255)	PRIMARY KEY	Foreign key tabel Exam_session field id
test_id	VARCHAR(255)	PRIMARY KEY	Foreign key tabel Test field id

status	INT(10)	NOT NULL	1: masih tersedia 0: tidak tersedia
createdAt	DATETIME	NOT NULL	tanggal data row pertama dibuat
updatedAt	DATETIME	NOT NULL	tanggal data row terakhir diperbarui

7. Tabel Section

Tabel Test merupakan sebuah tabel yang digunakan untuk menyimpan semua Section yang ada. Sebuah Test dapat memiliki banyak Section sehingga tabel ini memiliki hubungan one to many. Tabel ini memiliki relation ke tabel Test.

Tabel 3. 7
Tabel “Section”

Nama Field	Tipe	Costraint	Keterangan
id	BIGINT(20)	PRIMARY KEY	Auto increment
test_id	BIGINT(20)	NOT NULL	Menunjukkan section merupakan bagian Test mana Foreign key tabel Test field id
section_number	INT(20)	NOT NULL	Nomor urut section pada Test parent
instruction	VARCHAR(1000)	NOT NULL	Instruksi mengerjakan Section
duration	INT(10)	NOT NULL	Durasi pengerjaan Section dalam satuan menit
section_type	VARCHAR(255)	NOT NULL	Tipe soal di Section 1: essay 2: multiple choice
question_type	VARCHAR(255)	NOT NULL	Tipe pertanyaan di Section 1: text 2: image
option_num	INT(10)	NOT NULL	Jumlah pilihan pertanyaan terbanyak (kalau abcde berarti 5) Apabila esai diisi 0
status	INT(10)	NOT NULL	1: masih tersedia 0: tidak tersedia

createdAt	DATETIME	NOT NULL	tanggal data row pertama dibuat
updatedAt	DATETIME	NOT NULL	tanggal data row terakhir diperbarui

8. Tabel Question

Tabel Question merupakan sebuah tabel yang digunakan untuk menyimpan semua pertanyaan yang ada. Tabel ini memiliki foreign key ke tabel Section.

**Tabel 3.8
Tabel “Question”**

Nama Field	Tipe	Costraint	Keterangan
id	BIGINT(20)	PRIMARY KEY	Auto increment
instruction	VARCHAR(1000)	NOT NULL	field soal/perintah dari pertanyaan tersebut
section_id	BIGINT(20)	NOT NULL	Menunjukkan Question merupakan bagian Section mana Foreign key tabel Section field id
option_num	INT(10)	NOT NULL	Banyak pilihan jawaban yang ada di Question ini. Apabila tidak memiliki maka bernilai 0
option_a	VARCHAR(255)	NULL	Apabila pilihan ada lebih dari abcde, maka pilihan akan dimasukkan ke option_a dan dipisah dengan ‘,’
option_b	VARCHAR(255)	NULL	
option_c	VARCHAR(255)	NULL	
option_d	VARCHAR(255)	NULL	
answer	VARCHAR(255)	NOT NULL	Jawaban dari pertanyaan. Bisa a/b/c/d/e, bisa jawaban esai.
option_type	INT(10)	NOT NULL	Tipe pilihan jawaban 1: text, 2: image

instruction_type	INT(10)	NOT NULL	Tipe instruksi 1: text, 2: image
status	INT(10)	NOT NULL	1: masih tersedia 0: tidak tersedia
createdAt	DATETIME	NOT NULL	tanggal data row pertama dibuat
updatedAt	DATETIME	NOT NULL	tanggal data row terakhir diperbarui

9. Tabel Test_Result

Tabel Test_Result merupakan sebuah tabel yang digunakan untuk menyimpan hasil pengerjaan Test yang dilakukan oleh setiap Exam_Session. Tabel ini memiliki foreign key ke tabel Test dan Exam_Session.

Tabel 3.9
Tabel “Test Result”

Nama Field	Tipe	Costraint	Keterangan
id	BIGINT(20)	PRIMARY KEY	Auto increment
test_id	BIGINT(20)	NOT NULL	Foreign key yang merujuk Test dengan field id
exam_session	BIGINT(20)	NOT NULL	Foreign key yang merujuk Esxam_Session dengan field id
start_date	DATETIME	NOT NULL	Kapan tes mulai dikerjakan
finish_date	DATETIME	NOT NULL	Kapan tes selesai dikerjakan
result	LONGTEXT	NULL	Hasil tes keseluruhan yang dikerjakan. Bisa berupa norma-norma yang sudah dikalkulasi program.
status	INT(10)	NOT NULL	1: masih tersedia 0: tidak tersedia
createdAt	DATETIME	NOT NULL	tanggal data row pertama dibuat

updatedAt	DATETIME	NOT NULL	tanggal data row terakhir diperbarui
-----------	----------	----------	--------------------------------------

10. Tabel Section_Result

Mirip dengan Test_Result, tabel Section_Result merupakan sebuah tabel yang digunakan untuk menyimpan hasil penggerjaan setiap Section yang dilakukan oleh setiap Exam_Session. Tabel ini memiliki foreign key ke tabel Section, Test_Result dan Exam_Session.

**Tabel 3. 10
Tabel “Section_Result”**

Nama Field	Tipe	Costraint	Keterangan
id	BIGINT(20)	PRIMARY KEY	Auto increment
section_id	BIGINT(20)	NOT NULL	Foreign key yang merujuk Section dengan field id
test_result_id	BIGINT(20)	NOT NULL	Foreign key yang merujuk Test_Result dengan field id
exam_session	BIGINT(20)	NOT NULL	Foreign key yang merujuk Exam_Session dengan field id
start_date	DATETIME	NOT NULL	Kapan tes selesai dikerjakan
finish_date	DATETIME	NOT NULL	Hasil tes keseluruhan yang dikerjakan. Bisa berupa norma-norma yang sudah dikalkulasi program.
num_correct	INT(20)	NOT NULL	Jumlah jawaban benar pada Section Result ini
status	INT(10)	NOT NULL	1: masih tersedia 0: tidak tersedia
createdAt	DATETIME	NOT NULL	tanggal data row pertama dibuat
updatedAt	DATETIME	NOT NULL	tanggal data row terakhir diperbarui

11. Tabel Question_Result

Mirip dengan Test_Result dan Section_Result, tabel Question_Result merupakan sebuah tabel yang digunakan untuk menyimpan hasil penggerjaan setiap Question yang dilakukan oleh setiap Exam_Session. Tabel ini memiliki foreign key ke tabel Question, Section_Result.

**Tabel 3. 11
Tabel “Question_Result”**

Nama Field	Tipe	Costraint	Keterangan
id	BIGINT(20)	PRIMARY KEY	Auto increment
section_result_id	BIGINT(20)	NOT NULL	Foreign key yang merujuk Section_Result dengan field id
question_id	BIGINT(20)	NOT NULL	Foreign key yang merujuk Question dengan field id
answer	VARCHAR(100)	NOT NULL	Jawaban yang sudah dikerjakan pelamar
status_correct	INT(10)	NOT NULL	Penanda apakah jawaban user benar atau salah
status	INT(10)	NOT NULL	1: masih tersedia 0: tidak tersedia
createdAt	DATETIME	NOT NULL	tanggal data row pertama dibuat
updatedAt	DATETIME	NOT NULL	tanggal data row terakhir diperbarui

12. Tabel IST_Data

Tabel IST_Data merupakan sebuah tabel yang digunakan untuk menyimpan hasil data yang harus diisikan oleh setiap penggerja tes sebelum memulai mengerjakan Tes IST. Tabel ini memiliki foreign key ke tabel Test_Result dan Exam_Session.

**Tabel 3. 12
Tabel “IST_Data”**

Nama Field	Tipe	Costraint	Keterangan
-------------------	-------------	------------------	-------------------

test_result_id	BIGINT(10)	PRIMARY KEY	Auto increment
email	VARCHAR(1000)	NOT NULL	Foreign key yang merujuk Exam_Session dengan field id
tanggal_lahir	DATE	NOT NULL	
tanggal_tes	DATE	NOT NULL	Kapan pelamar mengerjakan Test
usia	INT(255)	NOT NULL	
tujuan_tes	VARCHAR(255)	NOT NULL	Alasan mengapa mengikuti tes
status	INT(10)	NOT NULL	1: masih tersedia 0: tidak tersedia
createdAt	DATETIME	NOT NULL	tanggal data row pertama dibuat
updatedAt	DATETIME	NOT NULL	tanggal data row terakhir diperbarui

13. Tabel Kreapelin_Data

Seperti IST_Data, tabel Kreapelin_Data merupakan sebuah tabel yang digunakan untuk menyimpan hasil data yang harus diisi oleh setiap peserta tes sebelum memulai mengerjakan Tes Kreapelin. Tabel ini memiliki foreign key ke tabel Section_Result dan Exam_Session.

Tabel 3. 13
Tabel “Kreapelin_Data”

Nama Field	Tipe	Costraint	Keterangan
section_result_id	BIGINT(10)	PRIMARY KEY	Foreign key yang merujuk Section_Result dengan field id
email	VARCHAR(100)	NOT NULL	Foreign key yang merujuk Exam_Session dengan field id
pendidikan	VARCHAR(100)	NOT NULL	Pendidikan yang ada adalah smea, stm-smk, sma, sarjana muda, dan sarjana
jurusran	VARCHAR(100)	NOT NULL	ipa atau ips
jenis_kelamin	VARCHAR(100)	NOT NULL	l: laki-laki

			p: perempuan
status	INT(10)	NOT NULL	1: masih tersedia 0: tidak tersedia
createdAt	DATETIME	NOT NULL	tanggal data row pertama dibuat
updatedAt	DATETIME	NOT NULL	tanggal data row terakhir diperbarui

14. Tabel EPPS_Data

Seperti IST_Data dan Kreapelin_Data, tabel EPPS_Data merupakan sebuah tabel yang digunakan untuk menyimpan hasil data yang harus diisikan oleh setiap peserta tes sebelum memulai mengerjakan Tes EPPS.. Tabel ini memiliki foreign key ke tabel Section_Result dan Exam_Session.

Tabel 3. 14
Tabel “EPPS_Data”

Nama Field	Tipe	Costraint	Keterangan
test_result_id	BIGINT(10)	PRIMARY KEY	Foreign key yang merujuk Test_Result dengan field id
jenis_kelamin	VARCHAR(255)	NOT NULL	l: laki-laki p: perempuan
pendidikan	VARCHAR(255)	NOT NULL	Pendidikan yang ada adalah smea, stm-smk, sma, sarjana muda, dan sarjana
kode_epps	VARCHAR(1)	NOT NULL	1: college student (men) 2: college student (women) 3: general adult (men) 4: general adult (women)
status	INT(10)	NOT NULL	1: masih tersedia 0: tidak tersedia
createdAt	DATETIME	NOT NULL	tanggal data row pertama dibuat
updatedAt	DATETIME	NOT NULL	tanggal data row terakhir diperbarui

15. Tabel Section_Ongoing

Tabel Section_Ongoing merupakan sebuah tabel pembantu untuk mengindikasikan ada atau tidaknya tes yang sedang berlangsung. Tabel ini memiliki foreign key ke tabel Section dan Exam_Session.

Tabel 3. 15
Tabel “Section_Ongoing”

Nama Field	Tipe	Costraint	Keterangan
id	BIGINT (10)	PRIMARY KEY	
section_id	BIGINT	NOT NULL	Foreign key yang merujuk Section dengan field id
exam_session_id	BIGINT	NOT NULL	Foreign key yang merujuk Exam_Session dengan field id
start_status	INT	NOT NULL	Menandakan section sedang dikerjakan atau tidak 0: not started 1: running 2: finished
temp_answers	LONGTEXT	NOT NULL	Menyimpan jawaban sementara dari section yang dikerjakan yang nantinya bisa diload apabila halaman ter-refresh. Setiap jaaban dipisah dengan ‘;’
start_time	DATETIME	NOT NULL	Kapan tes mulai dikerjakan
finish_time	DATETIME	NOT NULL	Kapan tes selesai dikerjakan
duration	INT	NOT NULL	Durasi pengkerjaan tes berdasarkan menit
status	INT(10)	NOT NULL	1: masih tersedia 0: tidak tersedia
createdAt	DATETIME	NOT NULL	tanggal data row pertama dibuat
updatedAt	DATETIME	NOT NULL	tanggal data row terakhir diperbarui

3.3 Desain User Interface

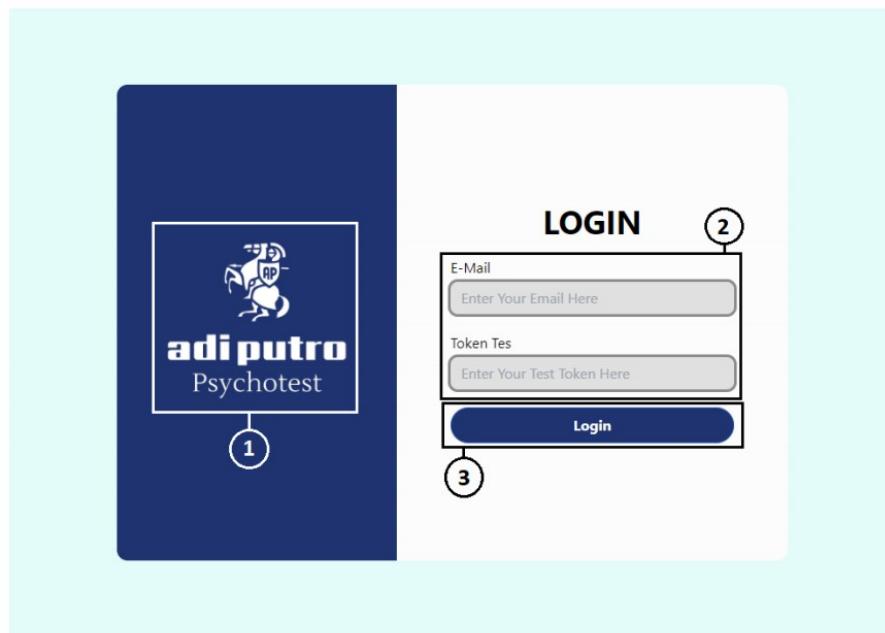
Desain Antarmuka Pengguna (User Interface Design) adalah desain untuk komputer, peralatan, mesin, perangkat komunikasi mobile, aplikasi perangkat lunak, dan situs web yang berfokus pada pengalaman pengguna (User Experience) dan interaksi. Tujuan dari Desain Antarmuka Pengguna adalah untuk membuat interaksi pengguna sesederhana dan seefisien mungkin, dalam hal mencapai tujuan pengguna atau apa yang sering disebut dengan user-centered design. Berikut ini akan dijelaskan lebih lanjut mengenai desain user interface dari website PT. Adiputro Wirasejati.

3.3.1 Halaman Job Applicant

Halaman Job Applicant adalah halaman yang dapat diakses oleh para pelamar kerja. Berikut akan dijelaskan bagian dari halaman job applicant.

1. Halaman Login

Halaman login adalah halaman pertama saat user mengunjungi website tes psikologi PT. Adiputro. Halaman ini berfungsi untuk melakukan login dan masuk ke dalam website sebagai peserta tes maupun sebagai admin. Tampilan dari halaman login dapat dilihat pada gambar 3.3.



Gambar 3.3
Halaman Login

Pada halaman ini, terdapat logo PT. Adiputro disebelah kiri (dapat dilihat pada gambar 3.3 nomor 1) dan form login di sebelah kanan (dapat dilihat pada gambar 3.3 nomor 2). Untuk bisa masuk ke dalam website, user harus mengisi email dan token tes ke dalam field yang sudah disediakan, email yang digunakan adalah email saat melamar pekerjaan sedangkan token tes akan dikirimkan melalui email. Setelah mengisi email dan token tes, user dapat menekan tombol login (dapat dilihat pada gambar 3.3 nomor 3) untuk masuk ke dalam halaman website masing-masing role

2. Halaman Apply Pekerjaan

Halaman apply pekerjaan adalah halaman dimana orang dapat melamar pekerjaan ke PT. Adiputro Wirasejati. Untuk masuk ke halaman ini, user dapat melakukan scan pada QR Code yang ada pada brosur lowongan kerja, lalu user akan dialihkan ke halaman apply pekerjaan. Tampilan halaman apply pekerjaan dapat dilihat pada gambar 3.4.

1

Biodata

NAMA LENGKAP : Nama Lengkap

USIA : 20

NO KTP : 0123456789

NO.TELEPON/HP : 081234567890

TEMPAT LAHIR : Malang

E-MAIL : contoh@example.com

TANGGAL LAHIR : hh/bb/tttt

JENIS KELAMIN : LAKI-LAKI PEREMPUAN

ALAMAT DOMISILI : Alamat Domisili

STATUS PERKAWINAN : BELUM MENIKAH SUDAH MENIKAH

PENDIDIKAN TERAKHIR :

GELAR :	Tingkat / Gelar
JURUSAN :	Jurusan
NAMA SEKOLAH :	Nama Sekolah / Universitas

2

POSIY YANG DILAMAR : Manager

3

PORTOFOLIO: Silahkan sertakan link portofolio anda (opsional):
Link Portofolio

4

LAMPIRAN: Silahkan upload lampiran file berekstensi .pdf
lampiran yang diupload (dihadkan 1 file):
Pilih File Tidak ada file yang dipilih

5

Kirim

Gambar 3.4
Halaman Apply Pekerjaan

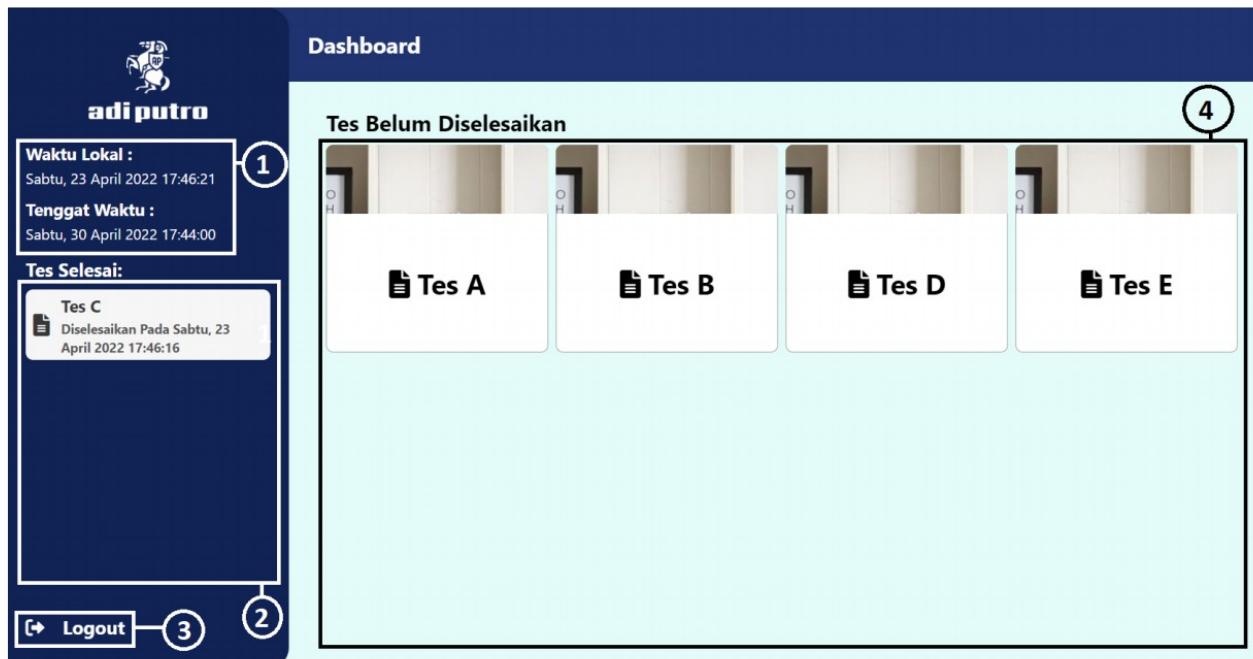
Untuk melamar pekerjaan, user harus mengisi biodata diri ke dalam field yang sudah disediakan (dapat dilihat pada gambar 3.4 nomor 1). User juga wajib mengisi posisi pekerjaan yang dilamar dengan memilih pada combobox posisi yang dilamar (dapat dilihat pada gambar 3.4 nomor 2). Combobox ini akan berisi semua posisi pekerjaan yang dibuka. Setelah mengisi posisi lamaran, user harus menyertakan link file portofolio bila ada (dapat dilihat pada gambar 3.4 nomor 3) dan mengupload berkas lampiran yang dibutuhkan seperti CV, surat lamaran, dll. (dapat dilihat pada gambar 3.4 nomor 4) dengan menekan tombol Choose File. Setelah semua sudah terisi, user dapat mengupload lamaran dengan menekan tombol kirim (dapat dilihat pada gambar 3.4 nomor 5).

3.3.2 Halaman Registrant

Halaman Registrant adalah halaman yang dapat diakses oleh user registrant. Berikut akan dijelaskan bagian dari halaman registrant.

1. Halaman Dashboard Registrant

Halaman dashboard registrant adalah halaman utama yang akan ditemui peserta tes saat setelah sukses melakukan login. Tampilan untuk halaman dashboard registrant dapat dilihat pada gambar 3.5.



Gambar 3.5
Halaman Dashboard Registrant

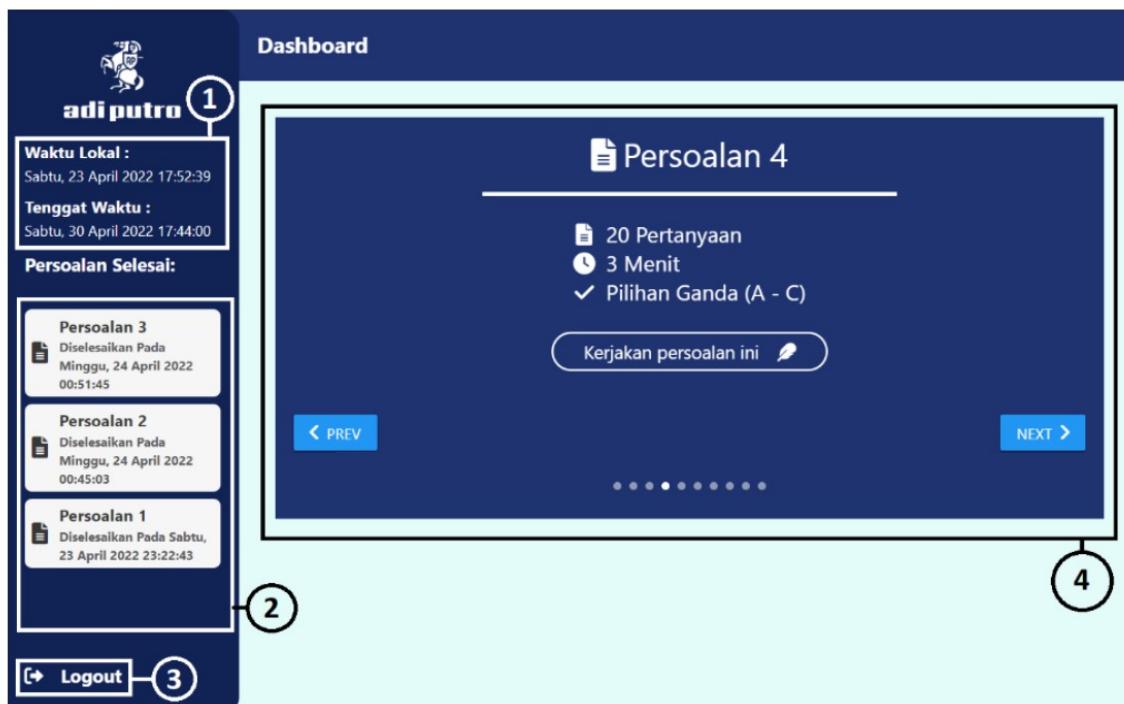
Pada halaman dashboard registrant, peserta tes dapat memilih tes yang akan dikerjakan terlebih dahulu. Terdapat beberapa komponen yang ada pada halaman ini dan akan dijelaskan urut sesuai nomor yang ada pada gambar 3.5, sbb:

1. Jam yang menunjukkan waktu lokal saat ini dan tenggat waktu atau batas peng交rajan tes.
2. Daftar tes yang sudah diselesaikan. Daftar ini akan ditampilkan urut dari yang paling baru diselesaikan sampai yang paling lama (dari atas ke bawah).
3. Tombol logout yang berfungsi untuk menghapus data temporary user yang sedang login dan mengarahkan user kembali ke halaman login.

4. Daftar tes yang belum diselesaikan oleh user. User dapat mengerjakan tes dengan menekan salah satu card/kotak tes, lalu user akan diarahkan menuju halaman dashboard persoalan yang berisi semua persoalan yang ada pada tes tersebut.

2. Halaman Daftar Persoalan

Halaman dashboard persoalan adalah halaman yang akan ditemui peserta tes saat setelah memilih tes yang ingin dikerjakan. Tampilan untuk halaman dashboard persoalan dapat dilihat pada gambar 3.6.



Gambar 3.6
Halaman Dashboard Persoalan

Pada halaman dashboard persoalan, peserta tes harus mengerjakan semua persoalan urut dari persoalan pertama sampai persoalan terakhir. Tiap tes memiliki jumlah persoalannya masing-masing, rentang jumlah persoalan tiap tes adalah 1 sampai 10. Terdapat beberapa komponen yang ada pada halaman ini dan akan dijelaskan urut sesuai nomor yang ada pada gambar 3.6, sbb:

1. Jam yang menunjukkan waktu lokal saat ini dan tenggat waktu atau batas pengerjaan tes.
2. Daftar persoalan yang sudah diselesaikan. Daftar ini akan ditampilkan urut dari yang paling baru diselesaikan sampai yang paling lama (dari atas ke bawah).
3. Tombol logout yang berfungsi untuk menghapus data temporary user yang sedang login dan mengarahkan user kembali ke halaman login.
4. Daftar persoalan yang belum diselesaikan oleh user. Terdapat tombol prev, next, dan kerjakan soal.

3. Halaman Pre-Test

Halaman pre-test adalah halaman yang akan menampilkan informasi cara mengerjakan persoalan. Tampilan untuk halaman dashboard persoalan dapat dilihat pada gambar 3.7.



Gambar 3.7
Halaman Pre-Test

Pada halaman dashboard persoalan, peserta tes harus mengerjakan semua persoalan urut dari persoalan pertama sampai persoalan terakhir. Tiap tes memiliki jumlah persoalannya masing-masing, rentang jumlah persoalan tiap tes adalah 1 sampai 10. Terdapat beberapa komponen yang ada pada halaman ini dan akan dijelaskan urut sesuai nomor yang ada pada gambar 3.7, sbb:

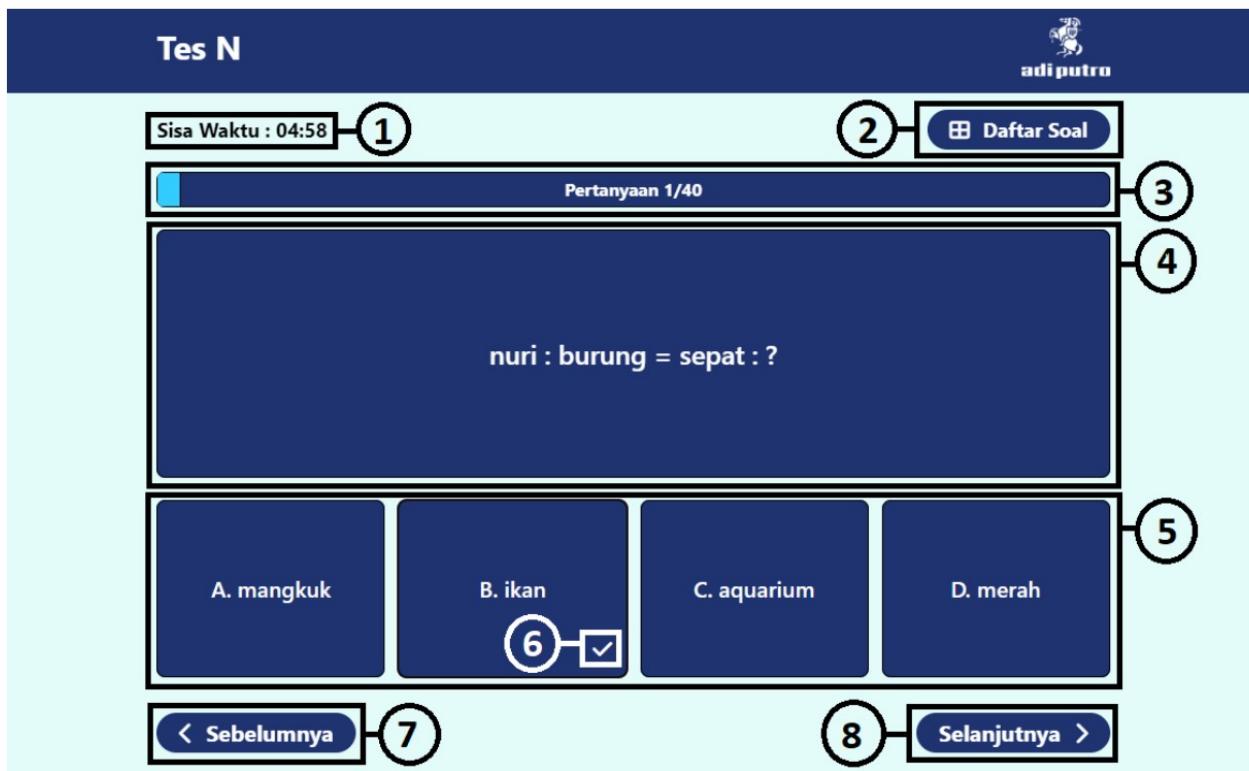
1. Tombol kembali berfungsi untuk mengarahkan user kembali ke halaman dashboard persoalan.
2. Kotak yang berisi informasi mengenai persoalan yang akan dikerjakan.
3. Bagian yang berisi instruksi cara mengerjakan persoalan, jumlah pertanyaan, waktu pengerjaan, jenis jawaban, dan simulasi berupa gambar gif (opsional).
4. Tombol mulai yang akan mengarahkan user ke halaman tes

4. Halaman Tes

Halaman tes adalah halaman yang berfungsi untuk mengerjakan persoalan yang diambil pada halaman dashboard persoalan. Terdapat 5 jenis tampilan tes yang ada pada website ini, sebagai berikut:

a. Halaman Tes Biasa

Halaman ini adalah halaman yang paling banyak digunakan oleh tes yang ada karena tampilan pada halaman ini adalah yang paling basic/dasar. Tampilan halaman tes biasa dapat dilihat pada gambar 3.8.



Gambar 3.8
Halaman Tes Biasa

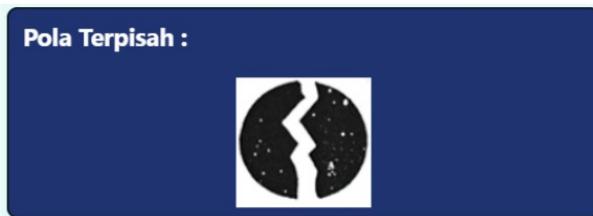
Pada halaman tes biasa, peserta tes dapat mengerjakan soal yang ada, apabila peserta tes tidak sempat menjawab semua persoalan maka jawaban yang dianggap valid hanya yang diisi. Terdapat beberapa komponen yang ada pada halaman ini dan akan dijelaskan urut sesuai nomor yang ada pada gambar 3.8, sbb:

1. Sisa waktu untuk mengerjakan persoalan
2. Tombol daftar soal yang akan membuka kotak daftar soal (dapat dilihat pada gambar 3.9). Pada kotak daftar soal, user dapat melihat soal mana saja yang sudah dikerjakan (ditandai dengan warna biru gelap), soal saat ini (ditandai dengan warna kuning), dan soal yang belum dikerjakan (ditandai dengan warna putih). User juga bisa lompat ke soal nomor-n dengan menekan kotak nomor-n.

Daftar Soal				
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	30
31	32	33	34	35

Gambar 3.9
Tampilan Daftar Nomor Soal

3. Progress bar yang menunjukkan nomor soal saat ini.
4. Kotak yang berisi soal yang bertipe teks. Selain teks, terdapat juga soal yang bertipe gambar (dapat dilihat pada gambar 3.10)

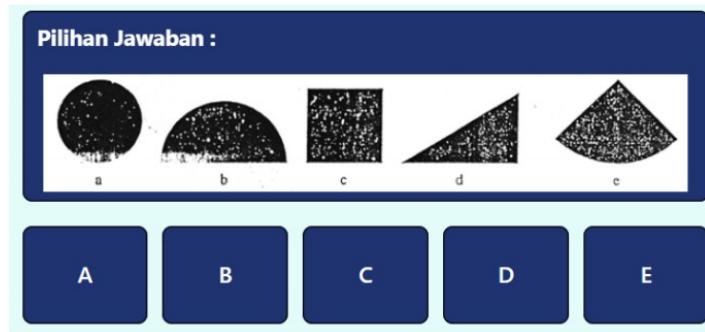


Gambar 3.10
Tampilan Pertanyaan Gambar

5. Kotak yang berisi pilihan jawaban yang bertipe pilihan ganda. Selain pilihan ganda, terdapat juga jawaban yang bertipe isian (dapat dilihat pada gambar 3.11) dan gambar (dapat dilihat pada gambar 3.12).

Jawab Di Sini

Gambar 3.11
Tampilan Jawaban Isian



Gambar 3.12
Tampilan Jawaban Gambar

6. Simbol centang menunjukkan jawaban yang dipilih oleh user
7. Tombol sebelumnya yang berfungsi mengarahkan user ke soal sebelumnya.
8. Tombol selanjutnya yang berfungsi mengarahkan user ke soal selanjutnya.

b. Halaman Tes EPPS/Papi Kostick

Halaman ini adalah halaman yang digunakan untuk tes EPPS dan tes Papi Kostick. Pada dasarnya halaman ini sama dengan halaman tes biasa, perbedaannya adalah tidak ada soal yang harus dijawab, peserta tes hanya perlu memilih pilihan A atau pilihan B yang menggambarkan dirinya. Pilihan ditampilkan 5 nomor untuk setiap halaman. Tampilan halaman tes EPPS/Papi Kostick dapat dilihat pada gambar 3.13.

Tes E

Sisa Waktu : 59:34

Halaman 1/45

Pilih salah satu pernyataan yang paling menggambarkan diri anda!

1.	A. opsi a ✓	B. opsi b
2.	A. opsi a	B. opsi b ✓
3.	A. opsi a ✓	B. opsi b
4.	A. opsi a	B. opsi b ✓
5.	A. opsi a ✓	B. opsi b

< Sebelumnya 1 Selanjutnya >

Gambar 3. 13
Halaman Tes EPPS/Papi Kostick

c. Halaman Tes Kraepelin

Halaman ini adalah halaman yang digunakan untuk tes kraepelin. Tampilan halaman tes Kraepelin dapat dilihat pada gambar 3.14.

Tes K

Untuk menjawab dapat menekan angka pada keyboard ataupun menekan tombol yang tersedia.

Sisa Waktu : 14 Detik 1 2

9 + 4 = ... 3 Berikutnya: 2 + 9 = ...

1	2	3
4	5	6
7	8	9
0		

Gambar 3. 14
Halaman Tes Kraepelin

Pada halaman tes kraepelin, peserta tes dapat mengerjakan soal hitungan yang ada dengan memasukkan angka satunya saja dengan menekan tombol yang ada pada gambar 3.14 nomor 4 ataupun menggunakan tombol pada keyboard (bila menggunakan laptop/pc), Terdapat beberapa komponen yang ada pada halaman ini dan akan dijelaskan urut sesuai nomor yang ada pada gambar 3.14, sbb:

1. Sisa waktu untuk mengerjakan persoalan pada kolom tersebut. Tiap kolom terdiri dari 27 soal, tugas user adalah mengerjakan sebanyak dan setepat mungkin.
2. Nomor yang menunjukkan nomor kolom saat ini
3. Soal yang berisi soal saat ini dan soal berikutnya
4. Keypad yang dapat digunakan untuk menjawab soal

d. Halaman Tes Ekspresi

Halaman ini adalah halaman yang digunakan untuk tes Ekspresi. Pada dasarnya halaman ini sama dengan halaman tes biasa, perbedaannya adalah jumlah jawaban tiap nomor adalah 2 (baik dan buruk). Peserta tes dapat memilih pilihan yang paling baik dengan menekan tombol O lalu menekan tombol jawaban, untuk memilih jawaban yang paling buruk, peserta tes dapat menekan tombol X lalu menekan tombol jawaban. Tombol X dan O dapat dilihat pada gambar 3.15 nomor 1, sedangkan jawaban yang dipilih ditunjukkan dengan simbol X dan O pada tombol jawaban yang dipilih (dapat dilihat pada gambar 3.15 nomor 2). Tampilan halaman tes Ekspresi dapat dilihat pada gambar 3.15.

Tes E



Sisa Waktu : 33:23

Daftar Soal

Pertanyaan 1/19

Klik lah tanda O dan pilihlah pernyataan mana yang baik. Klik lah tanda X dan pilihlah pernyataan yang jelek. Jadi nanti di tiap nomer akan ada 2 jawaban, yaitu satu lingkaran dan satu silang.

Ini Soal

X O 1

A. Pilihan A

B. Pilihan B

2

C. Pilihan C

< Sebelumnya

Selanjutnya >

Gambar 3. 15
Halaman Tes Ekspresi

e. Halaman Tes Gambar

Halaman ini adalah halaman yang digunakan untuk tes Gambar. Pada tes ini, peserta tes harus menggambar sesuai dengan instruksi yang telah diberikan (dapat dilihat pada gambar 3.16 nomor 1), lalu menguploadnya dengan menekan kotak upload gambar (dapat dilihat pada gambar 3.16 nomor 3) dan submit dengan menekan tombol submit (dapat dilihat pada gambar 3.16 nomor 4). User juga bisa kembali ke halaman sebelumnya dengan menekan tombol kembali (dapat dilihat pada gambar 3.16 nomor 2). Tampilan halaman tes Gambar dapat dilihat pada gambar 3.16.

Tes G

adiputro

Gambar lah manusia lengkap dari ujung kepala sampai ujung kakinya. Setelah digambar, silahkan gambarnya diberi nama, usia, jenis kelamin, dan aktivitasnya sedang apa. Setelah di gambar di kertas, silahkan di upload ke sini. NB: kertas folio dengan posisi kertas **PORTRAIT**. Gambar menggunakan **PENSIL**

1

2

3

4

Upload gambar anda (klik disini)

Submit >

Gambar 3. 16
Halaman Tes Gambar

5. Halaman Form Biodata

Halaman Form Biodata adalah halaman dimana peserta tes dapat mengisi data diri mereka. Untuk masuk ke halaman ini, user dapat menekan tombol biodata pada list tes yang belum dikerjakan. Tampilan halaman form biodata dapat dilihat pada gambar 3.17.



Biodata

[Kembali](#)
(1)
A

Biodata

1. NAMA LENGKAP :	9. AGAMA :
<input type="text" value="Nama Lengkap"/>	<input type="text" value="Agama"/>
2. NAMA PANGGILAN :	10. USIA :
<input type="text" value="Nama Panggilan"/>	<input type="text" value="20"/>
3. ALAMAT DI KTP :	11. TINGGI BADAN :
<input type="text" value="Alamat di KTP"/>	<input type="text" value="150"/>
4. ALAMAT SEKARANG :	12. BERAT BADAN :
<input type="text" value="Alamat Sekarang"/>	<input type="text" value="50"/>
5. TELEPON/HP :	13. NO KTP / SIM / PASPOR :
<input type="text" value="081234567890"/>	<input type="text" value="0123456789"/>
6. TEMPAT & TGL LAHIR :	14. GOLONGAN DARAH :
<input type="text" value="Malang / 22 Januari 2022"/>	<input type="text" value="A / B / AB / O"/>

(2)
B

Nama	Nama
<input type="text" value="Nama Lengkap"/>	<input type="text" value="Nama Lengkap"/>
Alamat	Alamat
<input type="text" value="Alamat"/>	<input type="text" value="Alamat"/>
No. Telepon	No. Telepon
<input type="text" value="081234567890"/>	<input type="text" value="081234567890"/>

Bawa segala biodata yang saya isi di atas adalah benar adanya.

(1)

(2)

(3)

Gambar 3. 17
(a) Form Biodata (atas) (b) Form Biodata (bawah)

Pada halaman ini terdapat tombol kembali yang berfungsi untuk mengarahkan user kembali ke halaman dashboard utama (dapat dilihat pada gambar 3.17 (a) nomor 1). User dapat mengisi biodata diri ke dalam field yang sudah disediakan (dapat dilihat pada gambar 3.17 (a) nomor 2). Pada bagian bawah, user harus mengisi kota asal saat mengisi biodata (dapat dilihat pada gambar 3.17 (b) nomor 1). User juga harus mengupload tanda tangan digital dengan menekan tombol Unggah Tanda Tangan

yang dapat dilihat pada gambar 3.17 (b) nomor 2. Setelah semua sudah terisi, user dapat mengupload biodata dengan menekan tombol kirim (dapat dilihat pada gambar 3.17 (b) nomor 3).

3.3.3 Halaman Admin

Halaman admin adalah halaman yang hanya dapat diakses oleh admin. Berikut akan dijelaskan bagian dari halaman admin.

1. Header Admin

Pada header admin terdapat tulisan “Hi, Admin” di sisi kiri (dapat dilihat pada gambar 3.18 nomor 1) dan waktu saat ini (dapat dilihat pada gambar 3.18 nomor 2) yang menunjukkan hari, tanggal, bulan, tahun, jam, menit, dan detik saat ini dan akan diupdate tiap detik. Tampilan header admin dapat dilihat pada gambar 3.18.



Gambar 3.18
Tampilan Header Admin

2. Navigation Bar Admin

Navigation bar admin adalah list menu yang berfungsi untuk navigasi antar halaman yang ada di halaman admin. Pada navigation bar admin terdapat logo PT. Adiputro di bagian atas (dapat dilihat pada gambar 3.19 nomor 1) serta 5 menu (dapat dilihat pada gambar 3.19 nomor 3). Untuk menunjukkan halaman yang sedang diakses, maka navigasi yang menunjukkan halaman saat ini akan diberikan warna yang lebih cerah (dapat dilihat pada gambar 3.19 nomor 2) yang selalu menunjuk ke halaman saat ini. Admin juga dapat melakukan logout dan kembali ke halaman login dengan menekan tombol Logout (dapat dilihat pada gambar 3.19 nomor 4). Tampilan navigation bar Admin dapat dilihat pada gambar 3.19.



Gambar 3.19
Navigation Bar Admin

3. Halaman Dashboard Admin

Pada halaman home admin, terdapat navigation bar yang ada di semua halaman admin (dapat dilihat pada gambar 3.20 nomor 1). Admin juga dapat mengakses tiap halaman master dengan menekan card-card master yang dapat dilihat pada gambar 3.20 nomor 2. Pada bagian bawah terdapat tabel yang menunjukkan 25 persoalan terakhir yang telah diselesaikan oleh peserta tes. Tampilan halaman dashboard admin dapat dilihat pada gambar 3.20.

Hi, Admin

Rabu, 01 Juni 2022 16:45:27

Home

1 **Master Pendaftar**
Master Pendaftar adalah halaman untuk melihat data hasil tes, biodata, dan laporan dari setiap pelamar kerja

2 **Master Tes**
Master Tes adalah halaman untuk mengelola (melihat, memperbarui, maupun menghapus) data tes, persoalan, dan pertanyaan

3 **Master Sesi**
Master Sesi adalah halaman untuk mengelola (melihat dan memperbarui) data sesi pengerjaan tes, termasuk tes yang diberikan

4 **Master Rekrutmen**
Master Rekrutmen adalah halaman untuk mengelola (melihat, memperbarui, maupun menghapus) data lowongan kerja dan pelamar kerja

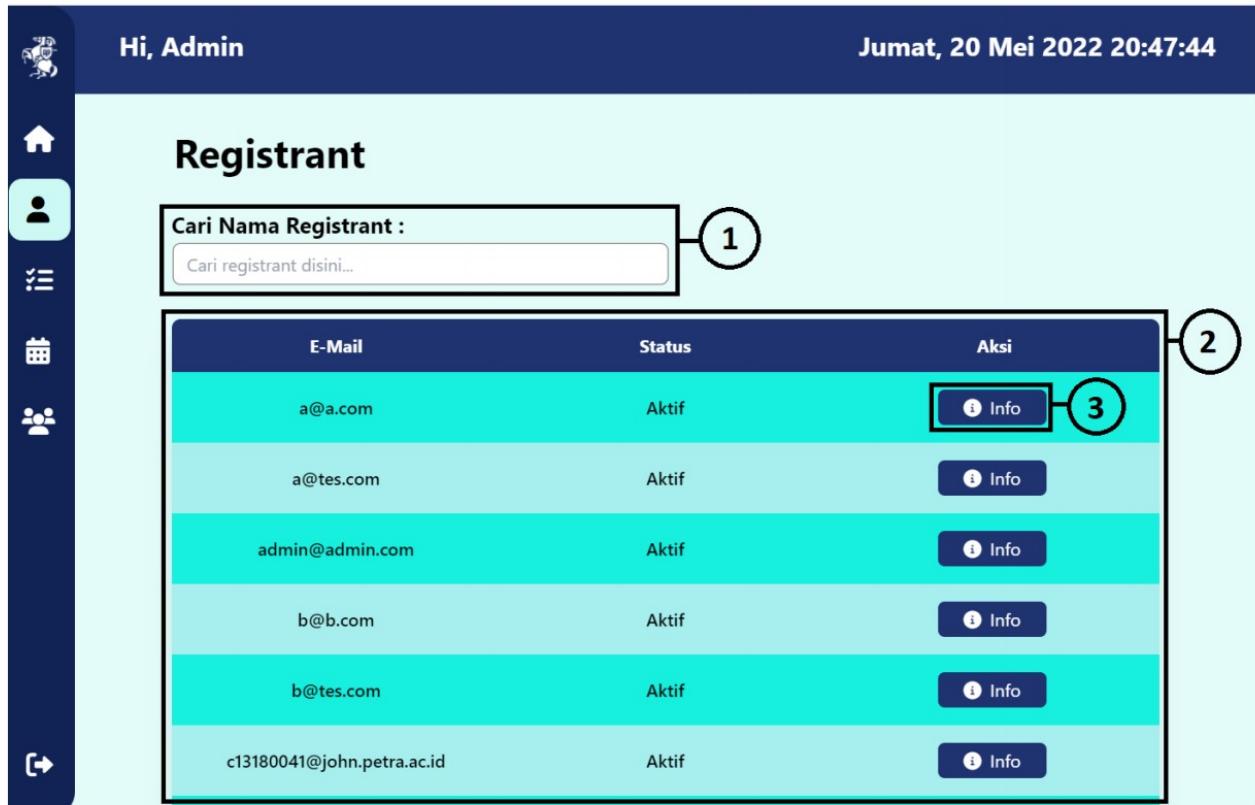
Persoalan yang baru saja diselesaikan

No.	E-Mail	Persoalan	Mulai	Selesai
1.	widean@tes.com	Tes Kreaplin Persoalan 1	01/06/2022 15:05	01/06/2022 15:08
2.	widean@tes.com	Tes Merakit Persoalan 1	01/06/2022 14:56	01/06/2022 14:56
3.	c@c.com	Tes Gambar Persoalan 1	06/04/2022 15:00	01/06/2022 14:00
4.	c@c.com	Tes Hafalan Persoalan 1	26/05/2022 15:05	26/05/2022 15:05
5.	c@c.com	Tes EPPS Persoalan 1	26/05/2022 14:54	26/05/2022 14:56
6.	c@c.com	Tes EPPS Persoalan 1	26/05/2022 14:54	26/05/2022 14:54
7.	c@c.com	Tes IST Persoalan 3	26/05/2022 06:46	26/05/2022 06:46
8.	c@c.com	Tes IST Persoalan 2	26/05/2022 06:46	26/05/2022 06:46
9.	c@c.com	Tes IST Persoalan 7	26/05/2022 06:46	26/05/2022 06:46
10.	c@c.com	Tes IST Persoalan 1	26/05/2022 06:46	26/05/2022 06:46
11.	c@c.com	Tes IST Persoalan 4	26/05/2022 06:46	26/05/2022 06:46

Gambar 3. 20
Halaman Dashboard Admin

4. Halaman Master Registrant

Pada halaman master registrant terdapat 2 sub halaman. Sub halaman yang pertama adalah halaman daftar registrant. Pada halaman ini terdapat search bar yang berfungsi mencari registrant berdasarkan email (dapat dilihat pada gambar 3.21 nomor 1) dan juga tabel list registrant yang menunjukkan detail dari setiap registrant yang sudah terdaftar (dapat dilihat pada gambar 3.21 nomor 2). Admin dapat melihat detail user lebih lanjut dengan menekan tombol Info yang ada pada kolom Aksi (dapat dilihat pada gambar 3.21 nomor 3). Tombol info ini akan mengarahkan admin ke halaman detail registrant. (dapat dilihat pada gambar 3.21)



Gambar 3. 21
Halaman Master Registrant

Sub halaman yang kedua adalah halaman detail registrant. Pada halaman ini ditampilkan detail hasil tes yang telah diselesaikan oleh peserta tes. Terdapat beberapa komponen yang ada pada halaman ini dan akan dijelaskan urut sesuai nomor yang ada pada gambar 3.22, sbb:

1. Tombol untuk melihat biodata yang telah diisi user
2. Combobox yang berisi daftar tes yang harus diselesaikan oleh peserta tes. Apabila admin memilih tes maka detail yang ada di bawah combobox juga akan ikut berubah
3. Detail hasil tes yang dipilih dari combobox
4. Daftar persoalan dari tes yang dipilih beserta hasil jawaban/nilai
5. Tombol reset yang berfungsi untuk mereset jawaban user sehingga user harus mengerjakan tes dari awal.

Hi, Admin

Kamis, 19 Mei 2022 19:31:06

Registrant Detail

1 Lihat Biodata

2 Tes IST

3

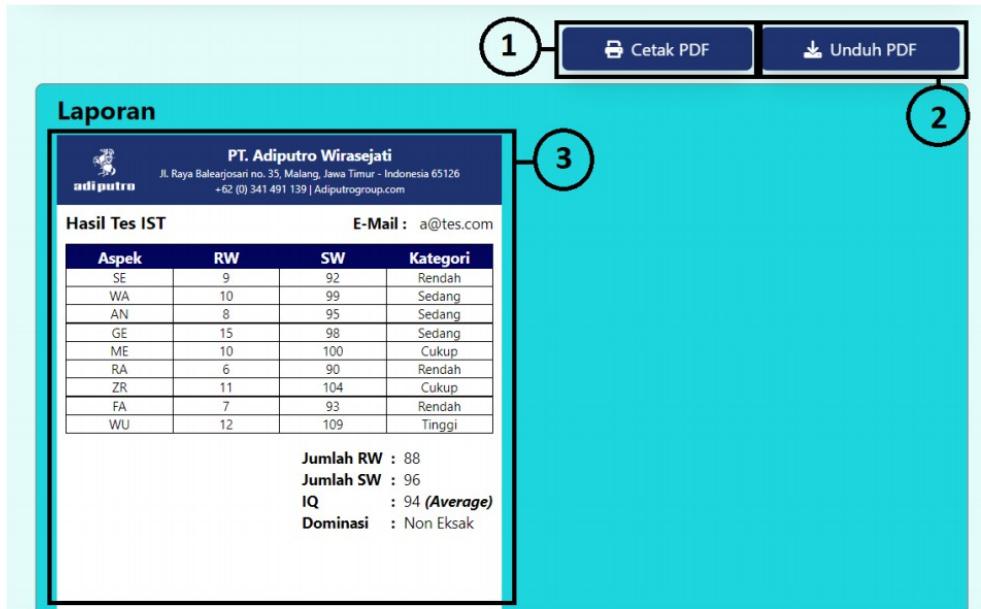
4

5

No. Seksi	Tipe Jawaban	Mulai	Selesai	Skor	Aksi
1	Pilihan Ganda	13/05/2022 21:31	13/05/2022 21:31	9/20	5 Reset
2	Pilihan Ganda	13/05/2022 21:32	13/05/2022 21:33	10/20	Reset
3	Pilihan Ganda	13/05/2022 21:33	13/05/2022 21:33	8/20	Reset
4	Esai	13/05/2022 21:33	13/05/2022 21:34	15/16	Reset
5	Esai	13/05/2022 21:34	13/05/2022 21:35	6/20	Reset

Gambar 3.22
Halaman Detail Registrant (atas)

Pada bagian bawah halaman detail registrant, terdapat report/laporan hasil tes dari tes yang dipilih (dapat dilihat pada gambar 3.23 nomor 3). Laporan ini dapat dicetak dengan menekan tombol Cetak PDF (dapat dilihat pada gambar 3.23 nomor 1) dan admin akan diarahkan ke halaman preview PDF, untuk mencetak admin dapat menekan tombol print. Admin juga dapat mengunduh PDF dengan menekan tombol Unduh PDF (dapat dilihat pada gambar 3.23 nomor 2). PDF yang akan dihasilkan adalah laporan gabungan semua tes yang sudah pernah dikerjakan oleh user.



Gambar 3.23
Halaman Detail Registrant (bawah)

Pada halaman biodata, admin dapat melihat semua data user yang telah dikirimkan user (dapat dilihat pada gambar 3.24 nomor 1). Halaman ini juga dilengkapi dengan tombol kembali yang akan mengarahkan user kembali ke halaman detail registrant (dapat dilihat pada gambar 3.24 nomor 2).

1. NAMA LENGKAP	: Lawrence Patrick Sianto	9. AGAMA	: kristen
2. NAMA PANGGILAN	: Lawrence	10. USIA	: 20 Tahun
3. ALAMAT DI KTP	: Jalan Jalan II	11. TINGGI BADAN	: 164 cm
4. ALAMAT SEKARANG	: Jalan Jalan III	12. BERAT BADAN	: 50.2 kg
5. TELEPON/HP	: 0812345678	13. NO KTP / SIM / PASPOR	: 981231515214561
6. TEMPAT & TGL LAHIR	: Surabaya, 01 Januari 1970	14. GOLONGAN DARAH	: O
7. KEWARGANEGARAAN	: WNI	15. NO NPWP	: 123456789
8. SUKU	: tionghoa	16. E-MAIL	: abc@def.com
17. Keadaan Kesehatan pada saat ini : baik/kurang sesuai Apakah anda pernah sakit keras/kecelakaan? ya/tidak, -/januari 2022		Apakah ada cacat tubuh? ya/tidak (Misal: penglihatan / pendengaran / pengucapan / jantung / dll)	
18. Apakah anda belum menikah/menikah? ya/tidak Nama Suami/Istri : perempuan laki Nama Anak : anak saya			

Gambar 3.24
Halaman Hasil Biodata

5. Halaman Master Sesi

Pada halaman master sesi, terdapat tombol Buat Sesi Baru yang berfungsi untuk membuka modal untuk membuat sesi tes untuk peserta tes (dapat dilihat pada gambar 3.25 nomor 2). Di bawah tombol Buat Sesi Baru terdapat daftar Sesi yang sudah terdaftar (dapat dilihat pada gambar 3.25 nomor 3). Admin dapat mencari sesi user dengan mengetikkan email user pada field Cari Sesi (dapat dilihat pada gambar 3.25 nomor 1). Sesi yang dibuat dapat diperbarui dengan menekan tombol perbarui pada baris sesi yang diinginkan (dapat dilihat pada gambar 3.25 nomor 4), tombol ini berfungsi untuk membuka modal untuk memperbarui sesi.

E-Mail	Mulai	Berakhir	Durasi	Token	Status	Aksi
widean2@yes.com	12/05/2022 19:21	13/05/2022 19:21	24 Jam	nRZNGnwmj89AkchMDiGow8	Aktif	Perbarui
a@tes.com	13/05/2022 19:17	25/05/2022 08:10	333.3 Jam	2x79Ty5kBV7tS1PM4Vuq9a	Aktif	Perbarui
b@tes.com	13/05/2022 19:17	25/05/2022 08:10	333.3 Jam	vdtzMyTMh4BTiQ7h61uT3s	Aktif	Perbarui
c@tes.com	13/05/2022 19:17	25/05/2022 08:10	333.3 Jam	6BFKeB448SgjanEWwwjCRb	Aktif	Perbarui
feybiola@tes.com	13/05/2022 19:17	25/05/2022 08:10	333.3 Jam	nz6VmQ1LnPH2qogWb2CGwo	Aktif	Perbarui
priskila@tes.com	13/05/2022 19:17	25/05/2022 08:10	333.3 Jam	7ff3xk7iCwqap1mLXEH3m	Aktif	Perbarui

**Gambar 3. 25
Halaman Master Sesi**

Pada modal Buat Sesi Baru (dapat dilihat pada gambar 3.26 (a)) terdapat tombol X yang berfungsi untuk menutup modal (dapat dilihat pada gambar 3.26 (a) nomor 1). Admin dapat membuat sesi untuk beberapa email sekaligus dengan mengisi field email registrant dan menekan tombol Tambahkan yang dapat dilihat pada gambar 3.26 (a) nomor 2. Tanggal tes dapat diatur dengan mengisi memilih tanggal dari dan sampai (dapat dilihat pada gambar 3.26 (a) nomor 3). Tes yang diberikan juga dapat dipilih dengan menekan tombol Paket Tes (dapat dilihat pada gambar 3.26 (a) nomor 4). Ada juga Tes Opsional yang dapat ditambahkan pada daftar tes yang diberikan.

Daftar tes yang akan diberikan dapat dihapus dengan menekan tombol X pada tes yang diinginkan. Untuk membuat sesi, admin dapat menekan tombol Buat (dapat dilihat pada gambar 3.26 (a) nomor 5). Modal Perbarui Sesi (dapat dilihat pada gambar 3.26 (b)) memiliki tampilan yang mirip dengan Modal Buat Sesi Baru, namun perbedaannya adalah admin dapat mengatur status aktif/non aktif dari sesi tersebut (dapat dilihat pada gambar 3.26 (b) nomor 2) dimana sesi yang nonaktif tidak dapat diakses oleh user. Admin juga dapat menambahkan tes dengan memilih tes pada combobox Tes Opsional lalu menekan tombol Tambahkan. Hanya tes yang baru saja ditambahkan saja yang dapat dibatalkan/dihapus dengan menekan tombol X pada tes tersebut (dapat dilihat pada gambar 3.26 (b) nomor 1). Untuk memperbarui data sesi, admin dapat menekan tombol Perbarui (dapat dilihat pada gambar 3.26 (b) nomor 3).

Gambar 3. 26
(a) Modal Sesi Baru (b) Modal Perbarui Sesi

6. Halaman Master Tes

Pada halaman master tes (bagian atas) terdapat beberapa komponen dan akan dijelaskan urut sesuai nomor yang ada pada gambar 3.27 sbb:

1. Combobox daftar nama tes, bila tes diganti maka list seksi yang ditampilkan akan ikut berubah pula.
2. Tombol Buat Seksi Baru yang berfungsi untuk membuka modal untuk membuat persoalan baru.
3. Daftar seksi/persoalan yang ada pada tes yang dipilih

4. Tombol detail yang bila ditekan akan menampilkan daftar pertanyaan yang ada pada persoalan tersebut (dapat dilihat pada gambar 3.27).
5. Tombol perbarui yang akan membuka modal perbarui seksi (tampilan sama dengan modal Buat Seksi Baru) yang berfungsi untuk memperbarui data seksi

The screenshot shows a user interface for managing test questions. At the top, it says "Hi, Admin" and "Kamis, 19 Mei 2022 12:50:55". On the left is a sidebar with icons for Home, Profile, Lists, Calendar, and Groups. The main area has a title "Tes" and a sub-section "Seksi". A form field "Nama Tes :" contains "Tes Tintum". Below is a table of questions:

Persoalan	Instruksi	Durasi	Tipe Jawaban	Tipe Pertanyaan	Jumlah Jawaban	Aksi
1	Di persoalan pertama sudah terdapat 10 nomer. Tugas an...	10 Menit	Teks	Esai	1	1 Detail 2 Perbarui
2	Untuk persoalan berikutnya. sudah terdapat 40 soal. Di ti...	7 Menit	Teks	Pilihan Ganda	4	1 Detail 2 Perbarui
3	Di persoalan ke 3 sudah terdapat 40 soal. Di tiap s...	5 Menit	Teks	Pilihan Ganda	4	1 Detail 2 Perbarui
4	Di persoalan berikutnya, sudah terdapat 20 soal. Di tiap s...	3 Menit	Teks	Pilihan Ganda	3	1 Detail 2 Perbarui
5	Di persoalan ke 5 merupakan soal cerita hitungan. Na...	7 Menit	Teks	Esai	1	1 Detail 2 Perbarui
6	Sudah terdapat 30 soal deret hitung. Dimana setiap s...	10 Menit	Teks	Esai	2	1 Detail 2 Perbarui
7	Di persoalan berikutnya sudah terdapat 20 soal. Di setiap s...	5 Menit	Teks	Pilihan Ganda	5	1 Detail 2 Perbarui
8	Di tiap nomer sudah terdapat 5	6 Menit	Teks	Pilihan Ganda	5	1 Detail 2 Perbarui

Annotations with numbers:

- 1: Next to the "Nama Tes :" input field.
- 2: Next to the "Buat Seksi Baru" button.
- 3: Next to the "Perbarui" button for question 1.
- 4: Next to the "Detail" button for question 1.
- 5: Next to the "Perbarui" button for question 5.

Gambar 3. 27
Halaman Master Tes (atas)

Pada halaman master tes (bagian bawah) terdapat beberapa komponen dan akan dijelaskan urut sesuai nomor yang ada pada gambar 3.28, sbb:

1. Tombol Buat Pertanyaan Baru yang berfungsi untuk membuka modal untuk membuat pertanyaan baru.
2. Tombol Hapus Semua Pertanyaan yang berfungsi mereset/menghapus semua pertanyaan yang ada pada seksi yang dipilih. Tombol ini hanya akan muncul pada seksi yang belum pernah dikerjakan sama sekali.
3. Daftar pertanyaan yang ada pada seksi/persoalan yang dipilih
4. Tombol Perbarui yang berfungsi untuk membuka modal Perbarui Pertanyaan

5. Tombol hapus yang berfungsi untuk menghapus pertanyaan tersebut dari seksi yang dipilih. Tombol ini hanya akan muncul pada pertanyaan yang belum pernah dijawab sama sekali.

No	Pertanyaan	Pilihan Jawaban	Kunci Jawaban	Aksi
1	Safir adalah ...	A. Pengembara, B. Gurun, C. Ahli filsafat, D. Batu-batuan	B	Perbarui Hapus
2	Safir adalah ...	A. Pengembara, B. Gurun, C. Ahli filsafat, D. Batu-batuan	B	Perbarui Hapus
3	Safir adalah ...	A. Pengembara, B. Gurun, C. Ahli filsafat, D. Batu-batuan	C	Perbarui Hapus
4	Safir adalah ...	A. Pengembara, B. Gurun, C. Ahli filsafat, D. Batu-batuan	B	Perbarui Hapus
5	Safir adalah ...	A. Pengembara, B. Gurun, C. Ahli filsafat, D. Batu-batuan	B	Perbarui Hapus
6	Safir adalah ...	A. Pengembara, B. Gurun, C. Ahli filsafat, D. Batu-batuan	C	Perbarui Hapus
7	Safir adalah ...	A. Pengembara, B. Gurun, C. Ahli filsafat, D. Batu-batuan	C	Perbarui Hapus

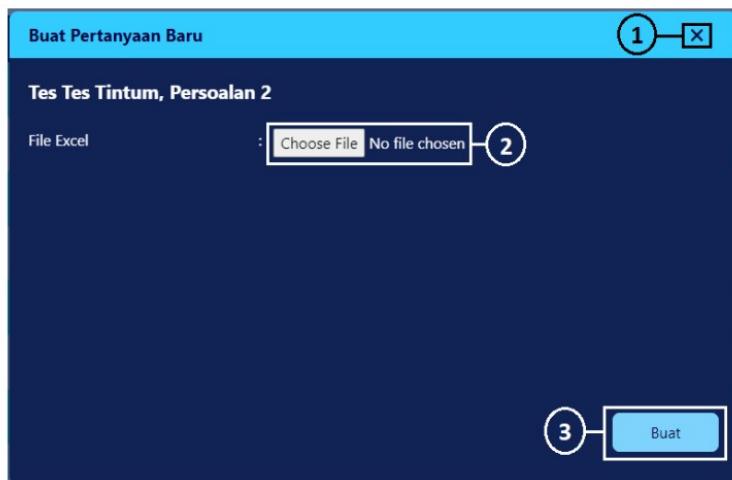
Gambar 3.28
Halaman Master Tes (bawah)

Pada modal Buat Seksi Baru terdapat tombol X yang berfungsi untuk menutup modal (dapat dilihat pada gambar 3.29 nomor 1). Pada modal ini admin dapat membuat seksi/persoalan baru dengan mengisikan data seksi pada field yang telah disediakan (dapat dilihat pada gambar 3.29 nomor 2) lalu menekan tombol Buat (dapat dilihat pada gambar 3.29 nomor 3).



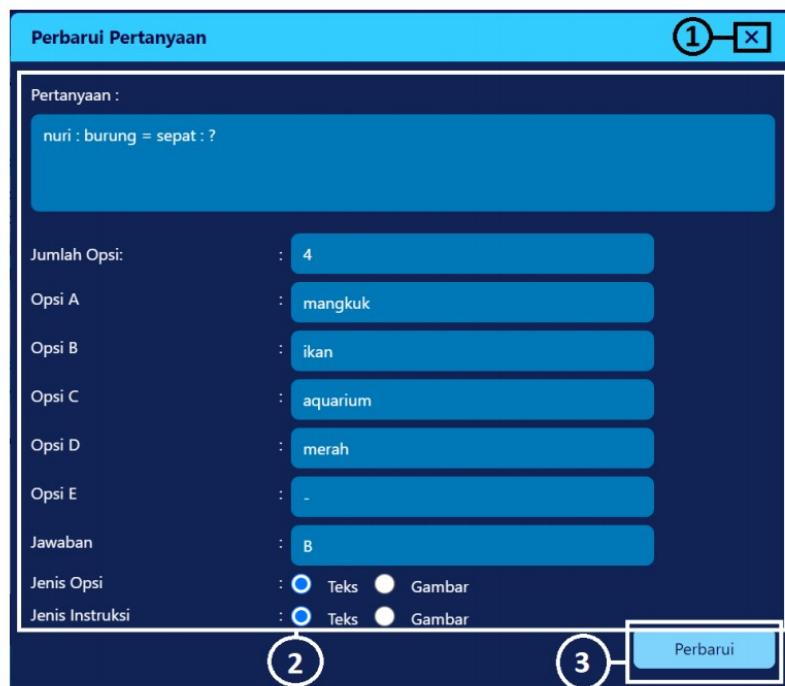
Gambar 3.29
Tampilan Modal Seksi Baru

Pada modal Buat Pertanyaan Baru terdapat tombol X yang berfungsi untuk menutup modal (dapat dilihat pada gambar 3.30 nomor 1). Pada modal ini admin dapat membuat pertanyaan baru dengan mengupload file excel pada field yang telah disediakan (dapat dilihat pada gambar 3.30 nomor 2) lalu menekan tombol Buat (dapat dilihat pada gambar 3.30 nomor 3). Pembuatan pertanyaan dilakukan secara langsung sehingga semua pertanyaan yang ada pada file excel dengan format yang ada akan masuk ke database.



Gambar 3.30 Tampilan Modal Pertanyaan Baru

Pada modal Perbarui Pertanyaan terdapat tombol X yang berfungsi untuk menutup modal (dapat dilihat pada gambar 3.31 nomor 1). Pada modal ini admin dapat memperbarui pertanyaan dengan mengisi data pertanyaan yang ingin diganti pada field yang telah disediakan (dapat dilihat pada gambar 3.31 nomor 2) lalu menekan tombol Perbarui (dapat dilihat pada gambar 3.31 nomor 3).



Gambar 3.31
Tampilan Modal Update Pertanyaan

7. Halaman Master Recruitment

Pada halaman master recruitment terdapat 2 sub halaman. Sub halaman yang pertama adalah halaman daftar lowongan pekerjaan (dapat dilihat pada gambar 3.32 (a)). Pada halaman ini terdapat tombol Buat Rekrutmen Baru yang berfungsi untuk membuat rekrutmen/lowongan baru (dapat dilihat pada gambar 3.32 (a) nomor 1), bila tombol ini ditekan maka akan ditampilkan modal untuk membuat rekrutmen baru (dapat dilihat pada gambar 3.32). Di Bawah tombol Buat Rekrutmen Baru terdapat tabel daftar lowongan yang terdaftar (dapat dilihat pada gambar 3.32 (a) nomor 2). Pada

tiap baris daftar lowongan terdapat kolom aksi yang berisi tombol detail (dapat dilihat pada gambar 3.32 (a) nomor 3) yang berfungsi untuk menampilkan daftar pelamar yang sudah mengajukan lamaran (dapat dilihat pada gambar 3.32 (b) nomor 1), dan tombol perbarui (dapat dilihat pada gambar 3.32 (a) nomor 4) yang berfungsi untuk mengupdate data lowongan, termasuk menutup lowongan. Admin dapat melihat detail data pelamar dengan menekan tombol Detail yang ada pada kolom Aksi pada tabel daftar pelamar pekerjaan (dapat dilihat pada gambar 3.32 (b) nomor 2). Tombol Detail ini akan mengarahkan admin ke halaman detail pelamar (dapat dilihat pada gambar 3.32).

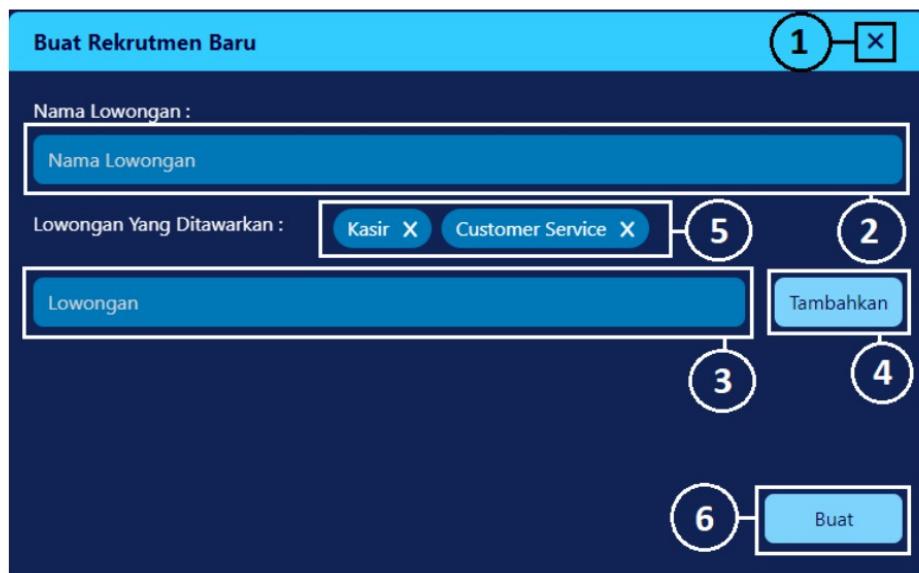
QR Barcode	Nama	Lowongan	Dibuat Pada	Status	Aksi
	Lamaran ini bagus	Manager,Kasir	18/03/2022 02:03	Open	Detail Perbarui
	Lamaran ini bagus	Manager,Kasir	18/03/2022 02:03	Open	Detail Perbarui

Email	Nama	Jenis Kelamin	Melamar Pada	Aksi
aa@gmail.com	Lawrence Patrick	Laki-Laki	17/03/2022 22:23	Detail
aaa@gmail.com	Lawrence Patrick	Laki-Laki	20/03/2022 10:55	Detail
aaaa@gmail.com	Lawrence Patrick	Laki-Laki	20/03/2022 11:37	Detail

Gambar 3.32
(a) Master Rekrutmen (atas) (b) Master Rekrutmen (bawah)

Pada modal pembuatan rekrutmen terbaru terdapat tombol X yang berfungsi untuk menutup modal (dapat dilihat pada gambar 3.33 nomor 1). Admin juga dapat

mengisikan nama lowongan yang dibuka pada field Nama Lowongan (dapat dilihat pada gambar 3.33 nomor 2). Selain itu, yang menjadi inti dari modal ini, lowongan yang dibuka dapat diisi pada field Lowongan (dapat dilihat pada gambar 3.33 nomor 3) dan menekan tombol Tambahkan (dapat dilihat pada gambar 3.33 nomor 4), maka secara otomatis lowongan akan bertambah pada daftar lowongan yang dapat dilihat pada gambar 3.33 nomor 5. Lowongan yang ditambahkan juga dapat dihapus dengan cara menekan tombol X pada lowongan yang ingin dihapus. Untuk membuat lowongan, admin dapat menekan tombol Buat (dapat dilihat pada gambar 3.33 nomor 6).



Gambar 3.33
Tampilan Modal Rekrutmen Baru

Sub halaman kedua adalah halaman detail pelamar. Pada halaman ini terdapat daftar data diri pelamar yang dapat dilihat pada gambar 3.34 nomor 1. Admin juga dapat melihat lampiran yang telah diupload oleh pelamar dengan menekan tulisan Lihat Lampiran (klik disini) (dapat dilihat pada gambar 3.34 nomor 2), sistem akan mengarahkan admin ke halaman file lampiran pada google drive.

Biodata

NAMA LENGKAP	:	Lawrence Patrick	NO.TELEPON/HP	:	08123456789
NO KTP	:	999999999	E-MAIL	:	aa@gmail.com
TEMPAT LAHIR	:	Surabaya	JENIS KELAMIN	:	Laki-Laki
TANGGAL LAHIR	:	2022-10-01	STATUS PERKAWINAN	:	Belum Menikah
ALAMAT DOMISILI	:	Jalan Pemangkas Rumput No 6	POSI SI YANG DILAMAR	:	Web Designer
USIA	:	20 Tahun	PENDIDIKAN TERAKHIR		
GELAR	:	SMA	LAMPIRAN : Lihat Lampiran (klik disini)		
JURUSAN	:	IPA	2		
NAMA SEKOLAH	:	SMA Ingin Berhasil			

Gambar 3.34
Halaman Detail Pelamar

BAB IV

IMPLEMENTASI

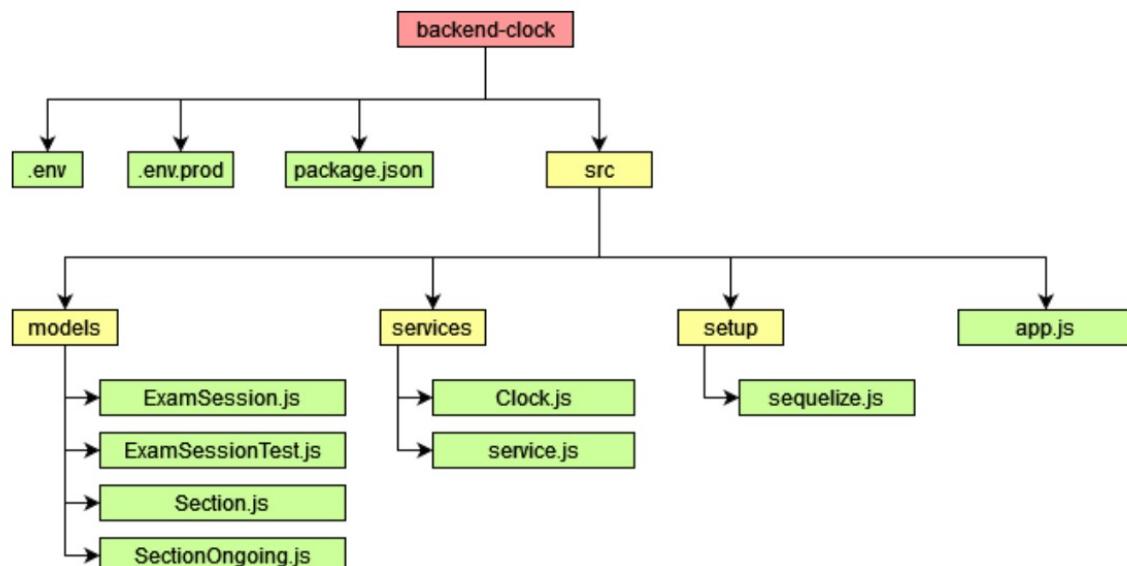
Setelah spesifikasi dan sistem telah ditentukan, maka tahap selanjutnya adalah mengimplementasikan ke dalam program. Pada bab ini akan dijelaskan mengenai implementasi tentang setiap fitur yang dimiliki oleh website tes psikologi PT. Adiputro Wirasejati.

4.1 Kerangka Program

Program yang dirancang dalam website ini akan terdapat 3 folder utama yang dipecah menjadi 3 yaitu backend-clock, backend, dan frontend. Setiap dari folder tersebut memiliki tujuannya masing-masing dan menggunakan arsitektur dan *library* yang berbeda. Berikut akan dijelaskan kerangka folder beserta file-file apa saja yang telah dibuat.

4.1.1 Backend-Clock

Bagian program yang pertama digunakan sebagai agen yang melakukan tick timer serta memantau apakah ada tes yang sedang dikerjakan. Program ini memanfaatkan package manager npm dengan javascript runtime node.js. Berikut adalah gambar struktur folder dan file.



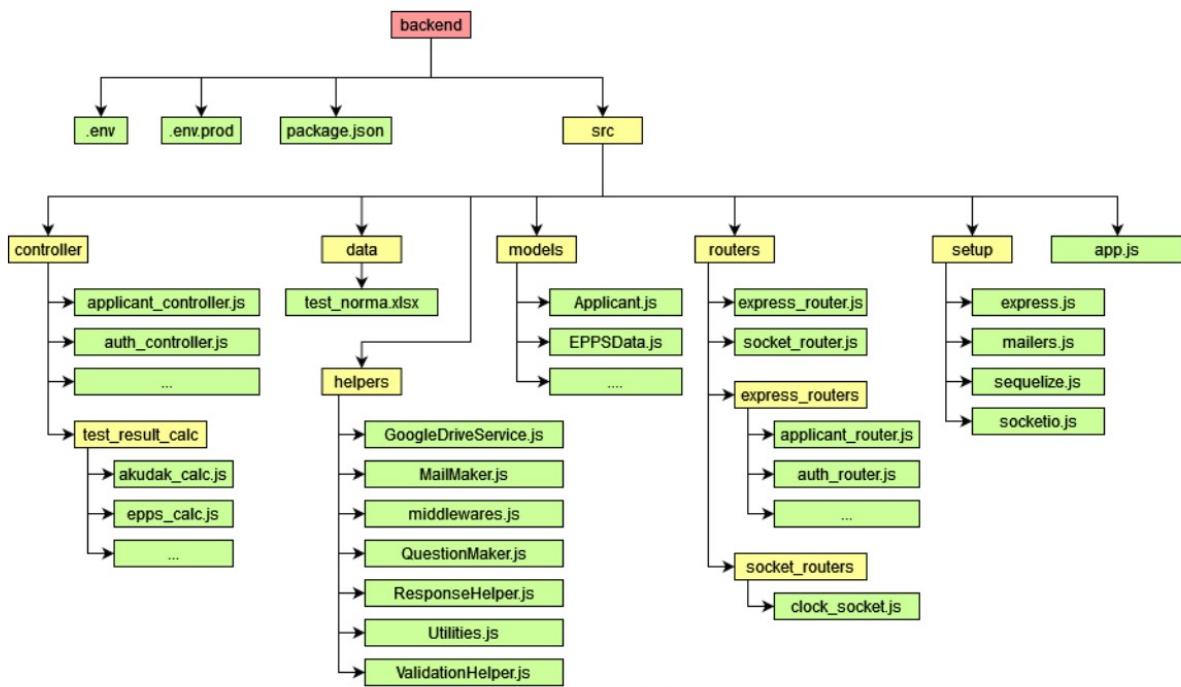
Gambar 4. 1
Struktur Folder Backend-Clock

Di dalam folder utama backend-clock terdapat 3 file dan 1 folder. Berikut akan dijelaskan kegunaan setiap folder.

- File .env digunakan untuk menaruh semua konfigurasi dari url server hingga secret token
- File .env.prod adalah file .env yang nantinya pada saat di deploy ke server, akan di copy dan di rename menjadi .env
- File package.json adalah file konfigurasi project backend-clock seperti metadata, dependency, dan juga script yang menggunakan Node.js.
- Folder src adalah folder utama yang menampung semua codingan yang ada. Di dalam folder ini terdapat beberapa file yaitu,
 - Folder Models yang berisi 4 model ExamSession.js, ExamSessionTest.js, Section.js, dan SectionOngoing.js. Model di sini berperan sebagai individu tabel yang ada di database.
 - Folder Services berisi Clock.js dan Service.js. File Clock berguna untuk membuat instance Clock untuk menghitung sisa waktu tes. Sedangkan file Service adalah file utama untuk memantau tes yang sedang dikerjakan serta memberi message ke backend apakah tes sudah selesai atau sedang berlangsung.
 - Folder Setup berisi file sequelize.js yang digunakan untuk inisialisasi library Sequelize untuk menghubungkan ke database.
 - File app.js adalah file yang memulai backend-clock agar bisa berjalan.

4.1.2 Backend

Bagian backend digunakan untuk mengatur alur data yang akan masuk dan keluar. Program ini juga memanfaatkan package manager npm dengan javascript runtime node.js. Berikut adalah gambar struktur folder dan file.



Gambar 4.2
Struktur Folder Backend

Di dalam folder backend terdapat 3 file dan sebuah folder src. Sama seperti di backend-clock, file .env, .env.prod, dan package.json memiliki fungsi yang sama. Dalam folder src, terdapat beberapa folder di dalamnya yaitu

- Folder Controller digunakan untuk menyimpan semua controller yang digunakan. Controller memiliki function-function yang digunakan untuk mengambil data, mengirim data, serta memproses data seperti kalkulasi norma. Berikut adalah semua isi dari folder controller.

Tabel 4.1
Tabel Folder Backend/Controller

applicant_controller.js	auth_controller.js
clock_controller.js	epps_data_controller.js
exam_session_controller.js	ist_data_controller.js
job_vacancy_controller.js	kreapelin_data_controller.js
question_controller.js	question_result_controller.js
registrant_controller.js	section_controller.js
section_ongoing_controller.js	section_result_controller.js

test_controller.js	Test_result_controller.js
--------------------	---------------------------

- Selain itu terdapat folder test_result_calc dimana semua file di dalamnya berguna untuk mengkalkulasi norma. Berikut isi dari folder ini.

Tabel 4. 2
Tabel Folder Backend/Controller/Test Result Calc

akudak_calc.js	epps_calc.js	hafalan_calc.js
ist_calc.js	kreapelin_cals.js	mmpi_calc.js
nine_calc.js	papi_kostick_calc.js	sdi_calc.js

- Folder Data digunakan untuk menyimpan file-file seperti excel sebagai panduan program dalam mengambil norma-norma yang nantinya akan dipakai sebagai kalkulasi tes. Hanya terdapat satu file yaitu test_norma.xlsx yang memiliki banyak sheet dinomori oleh kode tes.
- Folder Helpers digunakan untuk membantu program agar lebih terstruktur dan terbagi. Terdapat beberapa file yaitu

Tabel 4. 3
Tabel Folder Backend/Helpers

GoogleDriveService.js	MailerMaker.js
middlewares.js	QuestionMaker.js
ResponseHelper.js	Utilities.js
ValidationHelper.js	

- Folder Models berisi model-model yang setiap model berhubungan ke setiap tabel yang ada pada database. Terdapat beberapa file yaitu

Tabel 4. 4
Tabel Folder Backend/Models

Applicant.js	EPPSData.js	ExamSession.js
ExamSessionTest.js	ISTData.js	JobVacancy.js

KreapelinData.js	Question.js	QuestionResult.js
Registrant.js	Section.js	SectionOngoing.js
SectionResult.js	Test.js	TestResult.js

- Folder Routers berisi router dari API backend. Router digunakan untuk menentukan url route berupa GET, POST, atau yang lainnya. Setiap router yang ada pasti berhubungan ke satu controller pada folder Controller. Terdapat beberapa folder dan file yaitu
 - Terdapat folder express_routers yang berisi file express untuk API.

Tabel 4. 5
Tabel Folder Backend/Routers/Express Routers

applicant_route.js	auth_route.js
clock_route.js	epps_data_route.js
exam_session_route.js	ist_data_route.js
job_vacancy_route.js	kreapelin_data_route.js
question_result_route.js	question_route.js
registrant_route.js	section_ongoing_route.js
section_result_route.js	section_route.js
test_result_route.js	test_route.js

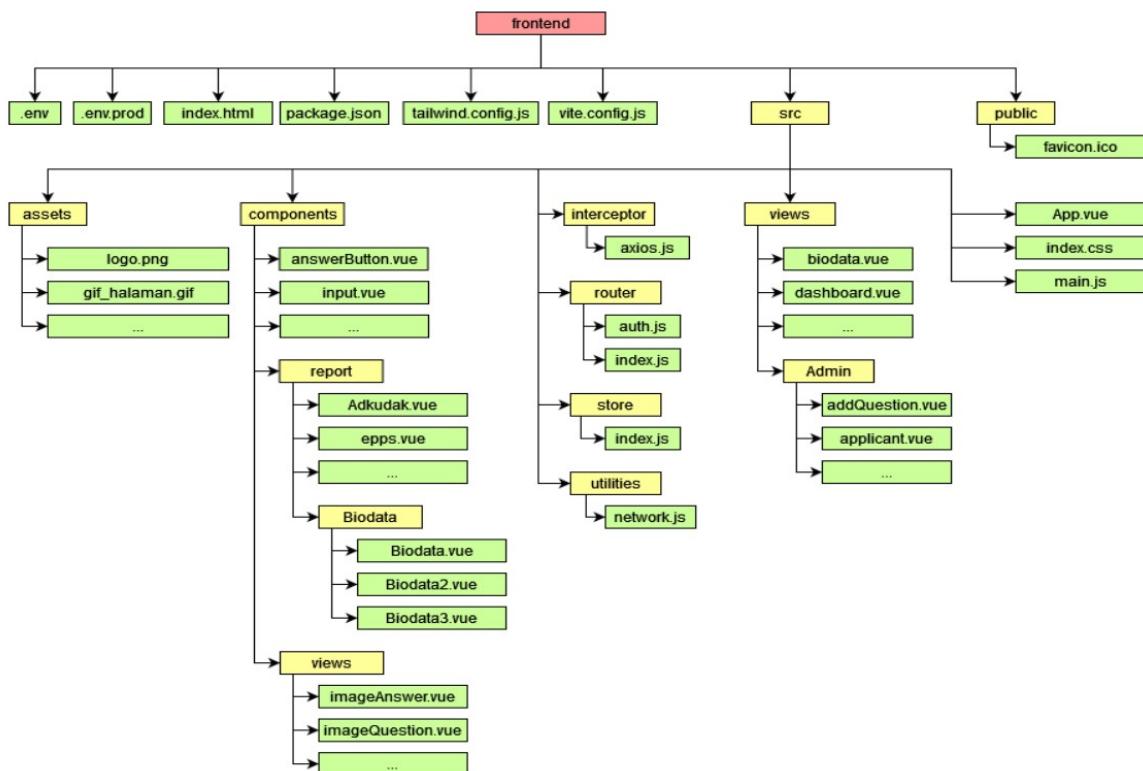
- Pada folder socket_routers hanya terdapat file clock_socket.js yang digunakan untuk koneksi socket ke frontend
- Selain itu terdapat 2 file express_router.js dan socket_router.js yang menginisialisasi dan menampung semua router express dan socket.
- Folder Setup digunakan untuk menyimpan file-file yang menginisialisasi backend mulai dari Express, Mailer, Sequelize, dan Socket. File-file yang ada yaitu:

express.js	mailer.js	sequelize.js	Socketio.js
------------	-----------	--------------	-------------

- File app.js adalah file yang memulai backend agar bisa berjalan.

4.1.3 Frontend

Bagian frontend digunakan untuk mengatur mengatur interface utama tes yang nantinya akan dilihat oleh pengguna. Pada frontend juga akan menggunakan package manager npm dengan framework Vue.js. Berikut adalah gambar struktur folder dan file.



Gambar 4. 3
Struktur Folder Frontend

Di dalam folder frontend terdapat 6 file, folder src, dan folder public. Sama seperti di backend-clock dan backend, file .env, .env.prod, dan package.json memiliki fungsi yang sama. Selain itu terdapat file index.html, tailwind.config.js, dan vite.config.js. Pada folder public, terdapat satu file ico bernama favicon.ico yang bertujuan untuk thumbnail website. Dalam folder src, terdapat beberapa folder di dalamnya yaitu

- Folder Assets menyimpan semua gambar, icon, atau animasi penunjuk yang digunakan pada web. Berikut adalah semua isi dari folder ini.

Tabel 4. 7
Tabel Folder Frontend/Assets

gif_deret_bilangan.gif	gif_hafalan.gif	gif_hafalan_2.gif
gif_hafalan_ist.gif	logo.png	logo2.png
logo_only.ico	logo_only.png	text_only.png

- Folder Components berisi komponen penyusun yang dapat dipakai berulang kali pada halaman web. Berikut adalah semua isi dari folder ini.

Tabel 4. 8
Tabel File Frontend/Components

answerButton.vue	input.vue	radiobutton.vue
table_input.vue	textarea.vue	

- Folder report yang menyimpan komponen untuk reporting. Terdapat beberapa file yaitu
 - Folder Biodata untuk komponen reporting biodata. Berisi 3 komponen yaitu Biodata.vue Biodata2.vue, dan Biodata3.vue

Tabel 4. 9
Tabel Folder Frontend/Components/Report

Adkudak.vue	epps.vue	eppsGraphics.vue
Hafalan.vue	IST.vue	ISTGraphic.vue
kecilkecil.vue	kraepelin.vue	kraepelinGraphics.vue
MMPI.vue	PapiKostick.vue	SDI.vue
tintum.vue	TintumAnak.vue	

- Folder views yang menjadi komponen besar sebuah halaman. Terdapat beberapa file yaitu

Tabel 4. 10
Tabel Folder Frontend/Components/Views

imageAnswer.vue	imageQuestion.vue	mChoiceAnswer.vue
Sections.vue	Test.vue	textAnswer.vue
textQuestion.vue		

- Folder Interceptor yang berisi satu file axios.js yang berguna untuk mendapatkan access token baru secara otomatis serta menjadi salah satu mekanisme autentikasi pengguna.
- Folder Router berguna untuk menyimpan semua router url dari halaman web. Hanya terdapat 2 file yaitu auth.js dan index.js.
- Folder Store berguna untuk mengatur access token dan refresh token. Hanya terdapat 1 file yaitu index.js.
- Folder Utilities berguna untuk membantu program pada website. Hanya terdapat 1 file yaitu network.js yang membantu inisialisasi socket ke backend.
- Folder Views menyimpan semua halaman pada website. Berikut adalah semua isi dari folder ini.

Tabel 4. 11
Tabel Folder Frontend/Views

biodata.vue	dashboard.vue	ekspresi.vue
epps.vue	gambarTest.vue	hafalan.vue
hafalanIST.vue	index.vue	kraepelin.vue
main_layout.vue	normalTest.vue	preExam.vue
recruitment.vue	section_dashboard.vue	

- Folder Admin menyimpan halaman yang khusus admin saja. Berikut adalah semua isi dari folder ini.

Tabel 4. 12
Tabel Folder Frontend/Views/Admin

applicant.vue	biodata.vue	dashboard.vue
master.vue	recruitment.vue	registrant.vue
registrantDetail.vue	reviewEssay.vue	session.vue

Test.vue		
----------	--	--

- File App.vue adalah file interface yang memulai frontend agar bisa berjalan.
- File index.css adalah file yang menyimpan semua styling dari website.
- File main.js adalah file yang memulai frontend agar bisa berjalan.

4.2 Implementasi Program

Pada website tes psikologi yang telah dikembangkan, terdapat 3 folder utama. Setiap folder program memiliki fungsinya masing-masing. Berikut akan dijelaskan lebih dalam mengenai kode yang ada di dalam ketiga folder tersebut:

4.2.1 Backend-Clock

Backend-Clock adalah sub-program dari program Backend yang berfungsi untuk menyediakan data waktu dari server. Program ini bertujuan agar semua data waktu yang digunakan pada website ini dapat disamakan berdasarkan jam yang ada pada server tempat program di-hosting. Berikut akan dijelaskan tentang program-program yang ada pada Backend-Clock:

1. Sequelize

Dalam menyambungkan program dengan database, program ini menggunakan package yang dibuat sebagai penyambung dari backend ke database. Package ini bernama Sequelize. Pada awal program, sequelize harus diinisialisasi agar bisa menggunakan service ini. Berikut adalah proses membuat sequelize yang terletak pada folder src/services/service.js.

Segmen Program 4.1 Sequelize

```

01: const db = new Sequelize(
02:   process.env.DB_NAME,
03:   process.env.DB_USER,
04:   process.env.DB_PASS,
05:   {
06:     host: process.env.DB_HOST,
07:     port: process.env.DB_PORT,
08:     dialect: "mysql",
09:     logging: false,
10:   }
11: );
12:

```

Segmen program 4.1 berisi code untuk mendeklarasi Sequelize. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 1 : program akan menginisialisasi class Sequelize baru.
- Baris 2-4 : program mengkonfigurasi Sequelize dengan nama, username, dan password database yang didapat dari file .env
- Baris 5-10 : program mengkonfigurasi server database berada di mana, port yang digunakan berapa, menggunakan DBMS apa, dan apakah query akan ditampilkan atau tidak.

2. Database Model

Model adalah sebuah class yang merepresentasikan sebuah tabel pada Database. Seperti yang sudah dijelaskan di Desain Database, terdapat 15 tabel pada database. Pada file-file model, hanya terdapat deklarasi atribut-atribut apa saja yang ada pada sebuah tabel tersebut. Berikut adalah salah satu contoh sebuah Model yang terletak pada folder src/models/ExamSessionTest.js.

Segmen Program 4.2 Database Model

```
01: export default class ExamSessionTest extends Model {}
02: ExamSessionTest.init(
03:   {
04:     exam_session_id: {
05:       type: DataTypes.BIGINT,
06:       primaryKey: true,
07:       allowNull: false,
08:     },
09:     test_id: {
10:       type: DataTypes.BIGINT,
11:       primaryKey: true,
12:       allowNull: false,
13:     },
14:     status: {
15:       type: DataTypes.INTEGER,
16:       allowNull: false,
17:       defaultValue: 1,
18:     },
19:   },
20:   {
21:     sequelize,
22:     modelName: "examsessiontest",
23:     tableName: "exam_session_test",
24:   }
25: );
26: ExamSessionTest.belongsTo(ExamSession, { foreignKey:
    "exam_session_id" });
```

Segmen program 4.2 berisi code untuk mendeklarasi sebuah Model. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 1 : program akan menginisialisasi class baru yang turunan dari class Model milik Sequelize.
- Baris 2 : program mulai mendeskripsikan atribut yang ada.
- Baris 4-8 : deklarasi atribut bernama ‘exam_session_id’ yang memiliki konfigurasi tipe data big integer, merupakan primary key, dan tidak null.
- Baris 14-18: deklarasi atribut bernama “status” yang memiliki konfigurasi tipe data integer, tidak boleh null, memiliki nilai default 1
- Baris 21: program meminta client Sequelize yang sudah dibuat untuk membuat model
- Baris 22 : program meminta nama modelName yang nantinya akan dipakai saat memanggil query.
- Baris 23 : program meminta nama tableName yang merupakan nama tabel yang dimaksud pada database
- Baris 26 : program membuat relation dari model ExamSessionTest ke Model ExamSession beserta foreign key nya.

3. Clock Tick

Pada backend-clock proses utama yang dilakukan hanyalah mengecek setiap detik apakah ada section tes yang sedang dikerjakan oleh user (exam_session). Apabila ada, maka backend-clock akan mengirimkan durasi dan sisa waktu tersisa ke backend lalu dari backend akan diteruskan ke frontend. Koneksi dari backend-clock ke backend menggunakan API POST. Berikut adalah proses tick backend-clock yang terletak pada folder src/services/service.js.

Segmen Program 4.3 Clock Tick

```
01: async function update() {
02:   console.log("Clock Ticking");
03:   SectionOngoing.findAll({where: {start_status:
04:     1}}).then((section_ongoing) => {
05:     if(section_ongoing.length <= 0) return;
06:     // Send Test Sessions to Update to Backend
07:     let data = [];
08:     for(let i = 0 ; i < section_ongoing.length; i++) {
          let countdown = Clock.second(new
Date(section_ongoing[i].finish_time), new Date());
```

```

09:
10:        if (countdown < 0){
11:            SectionOngoing.update({ start_status: 2 },
12:                { where: { id: section_ongoing[i].id}});
13:            continue;
14:        }
15:        data.push({
16:            "section_id": section_ongoing[i].section_id,
17:            "exam_session_id":
section_ongoing[i].exam_session_id,
18:            "countdown",
19:            "total_duration": section_ongoing[i].duration,
20:        })
21:    }
22:
23:    let token = jwt.sign({}, process.env.CLOCK_TOKEN, {expiresIn:
"10s",});
24:
25:    console.log('there are', data.length, 'tests running');
26:    post(process.env.BACKEND_URL + "/test/tick", data, {
27:        headers: {Authorization: `Bearer ${token}`},
28:    }).catch((error) => {
29:        console.log(error)
30:    });
31: }
32: }

```

Segmen program 4.3 berisi code untuk mengecek apakah ada section dan apabila ada maka clock akan mengirimkan data ke backend. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 3 : program akan mengambil semua section_ongoing yang masih start status 1.
- Baris 8 : program akan membuat class Clock yang memudahkan program menentukan waktu tersisa, durasi, serta section_id
- Baris 10-14 : program akan mengupdate status section_ongoing apabila durasi sudah melewati batas.
- Baris 23 : program membuat jwt token agar di backend, tidak sembarang request mengakses.
- Baris 26-30 : program mengirimkan data tiap section_ongoing ke backend dengan API POST beserta dengan jwt token yang ditaruh pada header.

4.2.2 Backend

Backend adalah program yang digunakan untuk menangani masalah pengolahan data seperti CRUD, authentication, perhitungan hasil tes, dan lain-lain. Program yang dibuat dapat menyediakan data dengan mengambil dari database dan diberikan kepada program frontend. Program backend ini juga dapat menampung dan mengolah hasil inputan user yang diberikan oleh program frontend. Berikut akan dijelaskan tentang program-program yang ada pada Backend:

1. Database Model

Model adalah sebuah class yang merepresentasikan sebuah tabel pada Database.

Seperti yang sudah dijelaskan di Desain Database, terdapat 15 tabel pada database.

Pada file-file model, hanya terdapat deklarasi atribut-atribut apa saja yang ada pada sebuah tabel tersebut. Berikut adalah salah satu contoh sebuah Model yang terletak pada folder src/models/Registrant.js.

Segmen Program 4.4 Database Model

```
01: class Registrant extends Model {}  
02: Registrant.init(  
03: {  
04:   email: {  
05:     type: DataTypes.STRING,  
06:     primaryKey: true,  
07:     allowNull: false,  
08:   },  
09:   /.../  
10:   status: {  
11:     type: DataTypes.INTEGER,  
12:     allowNull: false,  
13:     defaultValue: 1,  
14:     comment: "1: available, 0: deleted",  
15:   },  
16:   },  
17: {  
18:   sequelize,  
19:   modelName: "registrant",  
20:   tableName: "registrant",  
21: }  
22: );
```

Segmen program 4.4 berisi code untuk mendeklarasi sebuah Model. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 1 : program akan menginisialisasi class baru yang turunan dari class Model milik Sequelize.
- Baris 2 : program mulai mendeskripsikan atribut yang ada.

- Baris 4-8 : deklarasi atribut bernama ‘email’ yang memiliki konfigurasi tipe data String, merupakan primary key, dan tidak null.
- Baris 10-15 : deklarasi atribut bernama “status” yang memiliki konfigurasi tipe data integer, tidak boleh null, memiliki nilai default 1, dan memiliki comment.
- Baris 18 : program meminta client Sequelize yang sudah dibuat untuk membuat model
- Baris 19 : program meminta nama modelName yang nantinya akan dipakai saat memanggil query.
- Baris 20 : program meminta nama tableName yang merupakan nama tabel yang dimaksud pada database

2. Insert ke Database

Tentunya saat berhubungan dengan database, perlulah dilakukan insert data atau proses memasukkan data ke database. Dalam memasukkan data ke database, program memanfaatkan function yang sudah disediakan pada Model milik Sequelize. Berikut adalah salah satu contoh proses insert yang terletak pada folder src/controller/test_controller.js.

Segmen Program 4.5 Insert ke Database

```

01:  async create(req, res) {
02:    console.log("Creating A New Test...");
03:
04:    if (!validate_required_columns(req, Test, ["status"])) {
05:      missing_param_response(res);
06:      return;
07:    }
08:
09:    const new_test = await Test.create({
10:      name: req.body.name,
11:      test_order: req.body.test_order,
12:      test_type: req.body.test_type,
13:    });
14:
15:    success_response(res, new_test.toJSON(), "Create Successful!");
16:  }

```

Segmen program 4.5 berisi code untuk melakukan insert data ke database menggunakan Model. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 4-7 : program akan melakukan pengecekan inputan yang diberikan sudah lengkap atau tidak kecuali atribut status. Apabila tidak, maka program akan mengembalikan response gagal inputan kurang.
- Baris 9-13 : program akan melakukan insert ke database secara asynchronous dengan atribut-atribut yang ingin diinputkan. Atribut yang diinputkan bisa dari request body (didapat dari API) atau dari sumber lainnya. Apabila atribut tidak diikutsertakan maka akan diisi dengan default value.
- Baris 15 : program akan mengembalikan response berhasil dengan mengembalikan hasil data yang baru saja di insert.

3. Update ke Database

Selain insert ke, apabila ada data yang ingin diubah, kita melakukan proses update data ke database. Dalam update data ke database, program memanfaatkan function yang sudah disediakan pada Model milik Sequelize. Berikut adalah salah satu contoh proses update yang terletak pada folder src/controller/test_controller.js.

Segmen Program 4.6 Update ke Database

```

01:  async update(req, res) {
02:    console.log("Updating A Test...");
03:
04:    if (!validate_required_columns(req, Test, ["status"],
05:      ["updating_id"])) {
06:      missing_param_response(res);
07:      return;
08:    }
09:    Test.findOne({ where: { id: req.body.updating_id } }).then((test)
10:      => {
11:        if (!test) {
12:          data_not_found_response(res);
13:          return;
14:        }
15:        test.set({
16:          name: req.body.name,
17:          test_order: req.body.test_order,
18:          test_type: req.body.test_type,
19:        });
20:        test.save();
21:
22:        success_response(res, test?.toJSON(), "Update
successful!");
23:      });
24:    }

```

Segmen program 4.6 berisi code untuk melakukan update data ke database menggunakan Model. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 4-7 : program akan melakukan pengecekan inputan yang diberikan sudah lengkap atau tidak kecuali atribut status dan perlu atribut updating_id. Apabila tidak, maka program akan mengembalikan response gagal inputan kurang.
- Baris 9 : program akan mencari data pertama yang ingin diubah. Proses ini berlangsung asynchronous.
- Baris 10-13 : program akan mengecek apakah data yang dicari tersebut ada atau tidak di database. Apabila tidak ada maka akan mengembalikan response gagal data tidak ditemukan.
- Baris 15-19 : program akan mengganti data lama dengan data baru sesuai atribut yang ingin diganti.
- Baris 20 : program akan menyimpan data yang telah diubah ke dalam database.
- Baris 22 : program akan mengembalikan response berhasil dengan mengembalikan hasil data yang baru saja di update.

4. Delete dari Database

Selain insert dan delete, terkadang perlunya menghapus data yang tidak penting dari database. Dalam delete data ke database, program memanfaatkan function yang sudah disediakan pada Model milik Sequelize. Berikut adalah salah satu contoh proses delete yang terletak pada folder src/controller/question_controller.js.

Segmen Program 4.7 Delete dari Database

```
01:  async deletebyid(req, res) {
02:    console.log("Deleting Question by id...");
03:
04:    Question.findOne({ where: { id: req.params.question_id } }).then(
05:      async (data) => {
06:        if (!data) {
07:          data_not_found_response(res);
08:          return;
09:        }
10:
11:        await data.destroy();
12:      }
13:    );
14:  }
15:
16:  function data_not_found_response(res) {
17:    res.status(404).json({
18:      message: "Data not found"
19:    });
20:  }
21:
```

```

13:             success_response(
14:                 res,
15:                 "Delete by ID Successful!",
16:                 "Delete by ID Successful!"
17:             );
18:         }
19:     );
20: }

```

Segmen program 4.7 berisi code untuk melakukan delete data dari database menggunakan Model. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 4-5 : program akan mencari data pertama yang ingin dihapus. Proses ini berlangsung asynchronous.
- Baris 6-9 : program akan mengecek apakah data yang dicari tersebut ada atau tidak di database. Apabila tidak ada maka akan mengembalikan response gagal data tidak ditemukan.
- Baris 11 : program akan menghapus data yang sudah dipilih dari database.
- Baris 13-17 : program akan mengembalikan response dengan pesan “Delete by ID Successful!”.

5. Mengambil semua data di Database

Mengambil data dari database juga penting dimana proses ini adalah salah satu proses penting di backend. Sama seperti proses sebelum-sebelumnya, program memanfaatkan function yang sudah disediakan pada Model milik Sequelize untuk mengambil data. Berikut adalah salah satu contoh proses mengambil data yang terletak pada folder src/controller/applicant_controller.js.

Segmen Program 4.8 Mengambil data dari Database

```

01: async getAll(req, res) {
02:     console.log("Getting All Available Applicants...");
03:
04:     Applicant.findAll({ where: { status: 1 } }).then((applicants) => {
05:         if (applicants.length == 0) {
06:             data_not_found_response(res);
07:             return;
08:         }
09:
10:         success_response(res, applicants, "Get All Data
Successfull!");
11:     });
12: }

```

Segmen program 4.8 berisi code untuk mengambil semua data dari query dari database menggunakan Model. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 4 : program akan mencari semua data yang memiliki kriteria inputan. Proses ini berlangsung asynchronous.
- Baris 5-8 : program akan mengecek apakah data yang dicari tersebut ada atau tidak di database. Apabila tidak ada maka akan mengembalikan response gagal data tidak ditemukan.
- Baris 10 : program akan mengembalikan data hasil pencarian dan memberikan pesan sukses.

6. Upload ke Google Drive

Program ini juga memiliki fitur untuk menyimpan file yang memiliki format zip, gambar, ataupun file apapun ke dalam salah satu media penyimpanan terbesar yaitu Google Drive. Berikut adalah salah satu contoh proses upload gambar kertas buram yang terletak pada folder src/controller/test_controller.js.

Segmen Program 4.9 Upload ke Google Drive

```
01:     async uploadBuram(req, res) {  
02:     /.../  
03:  
04:     // Get Buram folder  
05:     const googleDriveService = new GoogleDriveService(  
06:         driveClientId,  
07:         driveClientSecret,  
08:         driveRedirectUri,  
09:         driveRefreshToken  
10:    );  
11:    let subfolders = await googleDriveService  
12:        .searchInParent(driveStorageID)  
13:        .catch((error) => {  
14:            console.error(error);  
15:            return null;  
16:        });  
17:    let subfolder = subfolders.filter(  
18:        (subfolder) => subfolder.name == "BURAM") [0];  
19:    let files = [];  
20:    for (let i = 0; i < req.files.length; i++) {  
21:        let uploadFile = req.files[i];  
22:        let ext = uploadFile.originalname.split(".");  
23:        ext = ext[ext.length - 1];  
24:        let file = await googleDriveService  
25:            .saveFile(  
26:                "Buram_" + req.body.email + "_" + (i + 1),
```

```

27:                     uploadFile.buffer,
28:                     uploadFile.mimetype,
29:                     subfolder.id
30:                 )
31:             .catch((error) => { console.error(error); });
32:         await googleDriveService
33:             .updatePermission(file.data.id, "reader",
34:                               "anyone")
35:             .catch((error) => { console.error(error); });
36:         files.push(
37:             "https://drive.google.com/file/d/" + file.data.id
38:             + "/view?usp=sharing"
39:         );
40:     return res.status(200).send(files);
41: }

```

Segmen program 4.9 berisi code untuk meng-upload file ke Google Drive serta mengganti permissionnya. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 5-10 : program akan membuat client Google Drive terlebih dahulu.
- Baris 11-16 : program mencari semua subfolder yang ada di folder drive berdasarkan drive ID yang sudah disediakan.
- Baris 17-18 : program akan mencari folder dengan nama “BURAM”
- Baris 22-23 : program akan mengambil file extension dari file.
- Baris 24-31 : program melakukan upload file ke database dengan format nama, buffer file, jenis tipe (mimetype) file, dan disimpan ke id subfolder yang mana. Apabila error akan muncul pesan error
- Baris 32-34 : program akan mengganti permission/izin akses siapa yang bisa mengakses file tersebut menjadi semua orang.
- Baris 36-38 : program menyimpan link drive dari hasil file yang sudah diupload tadi.
- Baris 40 : program mengembalikan semua link drive file.

7. Perhitungan Norma Tes

Pada backend, terdapat proses yang mana ditujukan untuk menghitung tes-tes dari hasil jawaban yang telah user inputkan. Hasil dari tes ini nantinya akan disimpan ke database tepatnya pada tabel Test_Result. Program akan menghitung norma-norma yang telah berbeda-beda setiap tes. Norma patokan setiap tes disimpan pada sebuah

file excel yang terletak pada src/data/test_norma.xlsx. Berikut adalah salah satu contoh proses perhitungan tes yang terletak pada folder src/controller/test_result_calc/nine_calc.js.

Segmen Program 4.10 Perhitungan Norma Tes

```
01:     async function calculate_nine_tests(test_type, testres, res) {
02:         SectionResult.findAll({ where: { test_result_id:
03:             testres.id } }).then(
04:             (sectionsres) => {
05:                 Section.findAll({
06:                     where: { test_id: testres.test_id },
07:                 }).then((sections) => {
08:                     let correct_data = [];
09:                     correct_data = correct_data.concat(
10:                         Array(sections.count).fill(0));
11:                     for (let i = 0; i < sections.count; i++) {
12:                         for (let j = 0; j < sectionsres.length;
13:                             j++) {
14:                             if (sectionsres[j].section_id ==
15:                                 sections.rows[i].id) {
16:                                 correct_data[sections.rows[i].section_number - 1] =
17:                                     sectionsres[j].num_correct;
18:                                 break;
19:                             }
20:                         }
21:                     // Calculate with norms
22:                     process_nine(
23:                         "./src/data/test_norma.xlsx",
24:                         test_type,
25:                         correct_data,
26:                         res,
27:                         testres
28:                     );
29:                 });
30:             );
31:         }
```

Segmen program 4.10 berisi code untuk kalkulasi hasil tes. Setiap jenis tes memiliki perhitungan berbeda. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2-6 : program akan mencari semua data yang diperlukan untuk menghitung tes. Data yang diperlukan adalah hasil jawaban dari user yang ada pada SectionResult dan Section. Perlu diingat bahwa tidak semua tes

mengambil dari SectionResult, bisa saja tes tersebut memerlukan jawaban asli dari setiap QuestionResult.

- Baris 7-18 : program akan menyimpan apakah data yang ingin dihitung benar atau salah.
- Baris 20-27 : program akan memanggil function yang bertujuan untuk melakukan kalkulasi program sesuai norma yang sudah ditentukan.

8. Socket

Pada website penggeraan tes ini, tentunya memiliki timer yang harus dikendalikan dari server. Pada arsitektur program kami, timer akan dijalankan dari server menggunakan backend-clock tetapi tidak menggunakan timer sungguhan. Melainkan backend-clock hanya mengirimkan waktu tersisa dari jam sekarang ke jam terakhir tes. Tugas backend adalah meneruskan request tersebut ke frontend menggunakan socket karena apabila menggunakan API, maka proses tidak akan berlangsung real-time. Berikut adalah proses mengirimkan socket ke frontend yang terletak pada folder src/controller/test_controller.js.

Segmen Program 4.11 Socket

```
01:  async tick(req, res) {
02:    // send countdown to frontend socket
03:    const clock_data = req.body;
04:    let userSocket = useUsersSocket();
05:    for (let i = 0; i < clock_data.length; i++) {
06:      for (let key in userSocket) {
07:        if (clock_data[i].exam_session_id == key) {
08:          // Send to each Corresponding Socket Client
09:          console.log("user socket", userSocket[key]);
10:          io.to(userSocket[key]).emit("test.tick", {
11:            countdown: clock_data[i].countdown,
12:            total_duration: clock_data[i].
13:              total_duration,
14:              section_id: clock_data[i].section_id,
15:            });
16:          break;
17:        }
18:      }
19:    }
20:    return res.status(200).send("OK");
}
```

Segmen program 4.11 berisi code mengirimkan socket ke frontend. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 3 : program menerima data id section beserta waktu tersisa.
- Baris 4 : program mengambil client socket yang sudah diinisialisasi pada awal program.
- Baris 5 : program mengirimkan semua tes yang sedang berlangsung ke client socket.
- Baris 6-7 : program mencari user socket id yang cocok dengan data dari backend-clock
- Baris 9-14 : program mengirimkan data section, total durasi, dan waktu tersisa ke user socket yang membutuhkan
- Baris 19 : program mengembalikan response berhasil

9. Mailer

Mailer adalah fitur untuk mengirim email secara otomatis ke email yang dituju. Hal yang dapat dikonfigurasi adalah subject, email pengirim, email tujuan, isi email, dan lain-lain. Berikut adalah fitur mengirimkan email yang terletak pada folder src/helpers/MailMaker.js.

Segmen Program 4.12 Mailer

```

01:  async function sendEmail({ receipients, subject, text = "", html =
02:    "" } = {}) {
03:    console.log(receipients);
04:    let config = {
05:      from: process.env.MAILER_SENDER, // sender address
06:      subject: subject, // Subject line
07:    };
08:    // Set Recipients
09:    let r = "";
10:    if (receipients.length == 1) {
11:      r = receipients[0];
12:      config.to = r; //receivers
13:    } else if (!Array.isArray(receipients)) {
14:      r = receipients;
15:      config.to = r; //receivers
16:    } else {
17:      for (let i = 0; i < receipients.length; i++) {
18:        r += receipients[i] + ",";
19:      }
20:      r = r.substring(0, r.length - 1);
21:      config.bcc = r; // list of receivers
22:    }
23:
24:    // Set other variables
25:    if (text != "") config.text = text;

```

```

26:     if (html != "") config.html = html;
27:
28:     // Send Email
29:     let mailer = await createMailer();
30:     let info = await mailer.sendMail(config);
31:
32:     console.log("Message sent: %s", info.messageId);
33: }

```

Segmen program 4.12 berisi code mengirimkan email. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 3-6 : program membuat konfigurasi email yaitu pengirim email dan subject email.
- Baris 9-22 : program menentukan ingin mengirim email ke siapa saja.
- Baris 25-26 : program akan membuat konfigurasi text dan html apabila menggunakan text atau html.
- Baris 29 : program membuat client Mailer yang sudah diinisialisasi pada awal program dijalankan.
- Baris 30 : program mengirim email berdasarkan variabel konfigurasi yang sudah dibuat sebelumnya.

4.2.3 Frontend

Frontend adalah program yang digunakan untuk menangani masalah interactive user interface. Program yang dibuat dapat memberikan tampilan website yang diakses oleh user. Program ini juga dapat membuat user berinteraksi dengan website seperti berpindah-pindah halaman, menginputkan data login, mengisi jawaban tes, mengisi form, mengupload gambar, dan lain-lain. Berikut akan dijelaskan tentang program-program yang ada pada Frontend:

1. Halaman Login

Halaman Login adalah halaman utama yang akan tampil pada saat orang mengakses website ini. Pada halaman ini nantinya user dapat memasukkan username dan password berupa email sebagai username dan test token sebagai password lalu menekan tombol login untuk melakukan login. Berikut akan dijelaskan tentang potongan program untuk melakukan login.

Segmen Program 4.13 Login

```

01: login(submitEvent) {
02:     let elements = submitEvent.target.elements;

```

```

03:      (async() => {
04:        axios.post('/auth/login', {email: elements.email.value,
05:          test_token: elements.test_token.value },
06:          { withCredentials: true })
07:        .then((e) => {
08:          if(e.response && e.response.status != 200) {
09:            Swal.fire({ title: e.response.data,
10:              .../ });
11:          }else{
12:            let {token, age} = e.data.refresh_token;
13:            let data_user = { "test": e.data.tests,
14:              "email": e.data.email,
15:              "exam_session": e.data.exam_session };
16:            this.$cookies.set('refresh_token', token, age);
17:            this.$cookies.set('data_registrant',
18:              JSON.stringify(data_user), age*10);
19:            localStorage.setItem('LS_ACCESS_KEY_VAR', `Bearer
20:              ${e.data.token}`);
21:            localStorage.setItem('LS_USER_KEY_VAR', `${token}`);
22:            if(e.data.is_admin) this.$router.push('/admin')
23:            else this.$router.push('/dashboard')
24:          });
25:      })

```

Segmen program 4.13 berisi code untuk melakukan login kedalam website. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2 : program mengambil isi dari form yang disubmit untuk dikirimkan kepada program backend.
- Baris 3-6 : program mengirimkan data login kepada program backend untuk diproses lebih lanjut
- Baris 8-10 : program melakukan pengecekan apakah hasil proses dari backend mengalami error atau tidak. Bila mengalami error maka program akan menampilkan pesan error, namun bila tidak maka program akan dilanjutkan ke baris program nomor 12.
- Baris 12-15 : program mengambil data hasil proses dari backend untuk digunakan.
- Baris 16-17 : program mengatur cookies untuk digunakan di dalam website sekaligus sebagai tanda kalau user sudah berhasil login.
- Baris 18-19 : program mengatur cookies refresh token untuk otorisasi web.

- Baris 20-21 : program melakukan pengecekan role user. Bila user merupakan admin maka akan dialihkan ke halaman admin, bila bukan (peserta tes) maka user akan dialihkan ke halaman dashboard peserta tes.

2. Halaman Admin

Halaman Admin adalah halaman yang hanya dapat diakses oleh admin website. Di dalam halaman admin terdapat beberapa sub halaman yang dapat diakses oleh admin saja. Berikut akan dijelaskan beberapa aksi yang dapat dilakukan oleh admin beserta potongan programnya.

a. Master Registrant

Master Registrant adalah fitur yang memungkinkan admin untuk melihat daftar semua peserta tes yang terdaftar. Admin juga dapat melihat detail hasil tes dari user tersebut. Selain itu admin juga dapat mencetak laporan hasil tes user dalam bentuk PDF. Apabila ternyata hasil jawaban peserta tes ada kekeliruan semisal peserta sudah memulai tes namun komputer/device mati dan waktu tes terlanjur habis, maka admin dapat mereset jawaban user yang mana membuat user dapat mengerjakan tes lagi. Untuk menghindari kekeliruan pengisian data saat mengerjakan tes kraepelin, admin dapat mengganti data diri user. Berikut adalah potongan program yang ada pada master registrant.

Segmen Program 4.14 Membuat PDF

```

01:  async makePDF(jenis){
02:      this.prints = true
03:      $('#spinner-modal').fadeIn("slow");
04:      this.$nextTick(async () => {
05:          window.html2canvas = html2canvas;
06:          /...
07:          let ctr = 0;
08:          let temp = document.querySelectorAll('.printPdf')
09:          let fullReport = []
10:          for (let i = 0; i < temp.length; i++)
11:          { fullReport.push(temp[i]) }
12:          for (let i = 0; i < fullReport.length; i++) {
13:              const report = fullReport[i];
14:              await html2canvas(report,{ "scale": 2 }).then(canvas => {
                var imgHeight = canvas.height * imgWidth /
                canvas.width;
15:                  var heightLeft = imgHeight;
16:                  let moreThan1 = false
17:                  var imgData = canvas.toDataURL('image/jpeg');
18:                  doc.addImage(imgData, 'JPEG', 0, position, imgWidth,
                imgHeight);

```

```

19:             heightLeft -= pageHeight;
20:             while (heightLeft >= 0) {
21:                 position += heightLeft - imgHeight - 4;
22:                 doc.addPage();
23:                 doc.addImage(imgData, 'PNG', 0, position,
24:                             imgWidth, imgHeight);
25:                 heightLeft -= pageHeight;
26:                 moreThan1 = true
27:             }
28:             if (!moreThan1) doc.addPage()
29:             ctr++
30:             if(ctr == fullReport.length) {
31:                 doc.deletePage (doc.internal.getNumberOfPages ())
32:                 $('#spinner-modal').fadeOut("slow");
33:                 if(jenis=='print')
34:                     window.open(doc.output('bloburl'), '_blank');
35:                 else if(jenis=='download')
36:                     doc.save(email+".pdf")
37:             }
38:         })
39:     }

```

Segmen program 4.14 berisi code untuk membuat laporan dalam bentuk PDF. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2 : program mengganti status print menjadi true untuk membuat semua dokumen yang ingin dijadikan PDF jadi terlihat/visible
- Baris 3 : program menampilkan animasi loading
- Baris 4 : program melakukan delay selama 1 detik untuk memastikan semua laporan sudah dirender dengan benar.
- Baris 5-7 : program mempersiapkan variabel-variabel yang dibutuhkan untuk proses pembuatan PDF
- Baris 8-10 : program mengambil semua laporan yang akan dijadikan PDF
- Baris 11-12 : program mengiterasi setiap laporan untuk dijadikan PDF satu persatu
- Baris 13 : program membuat laporan menjadi bentuk canvas
- Baris 14-16 : program menginisialisasi variabel

- Baris 17-18 : program mengkonversikan canvas menjadi gambar dengan ekstensi jpeg lalu ditambahkan kedalam dokumen PDF.
- Baris 19 : program mengurangi variabel heightLeft dengan pageHeight untuk melihat apakah masih ada sisa gambar yang belum dimasukkan PDF.
- Baris 20-26 : program menambahkan sisa gambar yang belum dimasukkan ke PDF selama masih ada sisa.
- Baris 27 : program melakukan pengecekan apabila jumlah halaman laporan tidak lebih dari 1, maka jumlah halaman PDF ditambahkan 1 untuk laporan berikutnya
- Baris 28-29: program menambahkan isi variabel counter/ctr sebanyak 1 setiap iterasi, lalu program melakukan pengecekan apabila semua laporan sudah dimasukkan kedalam PDF maka program akan dilanjutkan ke baris 30.
- Baris 30-31 : program menghapus halaman terakhir (blank) dan menghilangkan animasi loading.
- Baris 32-33 : program melakukan pengecekan apabila jenis yang diinginkan adalah print maka program akan membuka halaman baru (pdf viewer), sedangkan bila jenisnya adalah download maka program akan mengunduh pdf.
- Baris 37 : program mengganti status print menjadi false untuk menyembunyikan semua laporan yang telah dijadikan PDF.

Segmen Program 4.15 Ganti Data Kraepelin

```

01: submitKraepelinData(e) {
02:     let pendidikan = e.target["pendidikan"].value
03:     let jurusan = e.target["jurusan"].value
04:     let jk = e.target["jk"].value
05:     $('#spinner-modal').fadeIn("slow");
06:     axios.post(this.port+='/kreapelin_data/update',{
07:         "section_result_id": this.sectionResult[0].id,
08:         ...
09:         "jenis_kelamin": jk
10:     }).then((response) => {
11:         axios.post(this.port+='/test_result/calculateresult',{
12:             test_id: 5,
13:             email: this.email

```

```

14:         }).then((response) => {
15:             $('#spinner-modal').fadeOut("slow"),
16:             axios.get(` ${this.port}/section_result/getbytest
17: /5?email=${this.email}`)
18:             .then(({data}) => {
19:                 this.processSectionResult(data)
20:                 this.isKraepelin()
21:             })
22:         }).catch( error => { console.log('error: ' + error) });
23:     },

```

Segmen program 4.15 berisi code untuk mengganti data diri tes kraepelin.

Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2-4 : program mengambil isi dari form yang disubmit untuk dikirimkan kepada program backend.
- Baris 5 : program menampilkan animasi loading
- Baris 6-10 : program mengirimkan data baru untuk tes kraepelin kepada program backend untuk memperbarui data yang lama.
- Baris 11-14 : program mengirimkan data kepada program backend agar program backend dapat menghitung kembali norma dan hasil tes yang baru.
- Baris 15 : program menghilangkan animasi loading sebagai tanda bahwa proses sudah selesai.
- Baris 16-20 : program meminta data hasil tes yang baru dari program backend.
- Baris 22 : apabila terjadi error selama berkomunikasi dengan program backend, maka program akan menampilkan pesan error pada console.

Segmen Program 4.16 Reset Hasil Jawaban

```

01: resetHasil(sec_id){
02:     Swal.fire({ title: 'Yakin mereset hasil jawaban ini?',
03:         ... }).then((result) => {
04:         if (result.isConfirmed) {
05:             $('#spinner-modal').fadeIn("slow");
06:             axios.post(this.port+ '/question_result/resetQuestion', {
07:                 exam_session: this.exam_session,
08:                 section_id: sec_id
09:             }).then((response) => {
10:                 axios.get(this.port+ '/test_result/getbyemail/' +
11: this.$route.query.registrant)
12:                     .then(({data}) => (

```

```

12:             this.testResult = data,
13:             this.dataInit(),
14:             $('#spinner-modal').fadeOut("slow"),
15:             Swal.fire({ title: 'Sukses mereset jawaban!',
16:                         ...}),
17:             this.gantiTes(this.selectedTes)
18:         )));
19:     });
20:   });
21: );
22: }

```

Segmen program 4.16 berisi code untuk mereset hasil jawaban tes. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2-4 : program menampilkan pesan konfirmasi untuk mereset jawaban, apabila user setuju untuk reset maka program akan dilanjutkan pada baris 5
- Baris 5 : program menampilkan animasi loading
- Baris 6-9 : program mengirimkan data hasil jawaban persoalan yang ingin dihapus kepada program backend untuk dihapus.
- Baris 10-13 : program meminta hasil semua tes yang telah dikerjakan peserta tes guna memperbarui data yang tampil.
- Baris 14 : program menghilangkan animasi loading.
- Baris 15-16 : program menampilkan pesan sukses.
- Baris 17 : program memastikan tes yang sedang dipilih tidak berubah

b. Master Exam Session

Master Exam session adalah fitur yang memungkinkan admin untuk melihat daftar semua sesi tes yang terdaftar. Admin juga dapat membuat sesi baru dan memperbarui data sesi yang telah ada. Pada saat membuat sesi baru, admin dapat mengatur waktu penggeraan tes dan memilih tes apa saja yang akan diberikan pada peserta tes. Berikut adalah potongan program yang ada pada master exam session.

Segmen Program 4.17 Membuka Modal Update Sesi

```

01: async openModal(data2) {
02:   this.test_result = []
03:   this.current_test_list = []
04:   this.tests = []

```

```

05:     await
06:     axios.get(` ${this.port}/test/getbyExamSession/${data2.id}`)
07:     .then(({data}) => {
08:       if(data!=undefined){
09:         let idx = -1;
10:         for (let i = 0; i < this.list_tes.length; i++) {
11:           for (let j = 0; j < data.length; j++) {
12:             if (this.list_tes[i].id == data[j].test_id){
13:               this.tests.push(this.list_tes[i])
14:               this.current_test_list.push(this.list_tes[i])
15:               break;
16:             }
17:           }
18:           let temp = []
19:           for (let i = 0; i < this.list_tes.length; i++) {
20:             const tes = this.list_tes[i];
21:             if(!this.tests.includes(tes)){
22:               temp.push(tes)
23:             }
24:           }
25:           this.opsional = temp
26:           this.selectedOptional = this.opsional[0].id
27:         }
28:       }
29:     await
30:     axios.get(` ${this.port}/test_result/getbyemail/${data2.email}`)
31:     .then(({data}) => {
32:       if(data!=undefined){
33:         for (let i = 0; i < this.current_test_list.length; i++)
34:         {
35:           for (let j = 0; j < data.length; j++) {
36:             if (this.current_test_list[i].id ==
37: data[j].test_id){
38:               let t = data[i]
39:               t.status = t.status==1?true:false
40:               this.test_result.push(t)
41:               break;
42:             }
43:           }
44:         this.current_exam_session = data2.id
45:         /...
46:         this.headerModal = "Perbarui Sesi";
47:         $('#modalSession').fadeIn("slow");
48:         $('#bg').fadeIn("slow");
49:       }

```

Segmen program 4.17 berisi code untuk membuka modal update sesi. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2-4 : program mengosongkan array yang akan dipakai. `test_result` berfungsi untuk menentukan apakah tes yang diberikan telah selesai dikerjakan atau belum, `current_test_list` berisi data tes yang telah diberikan, sedangkan `tests` berisi daftar tes yang telah diberikan dan apabila ada tambahan tes maka akan masuk kedalam array ini.
- Baris 5-7 : program meminta data tes yang telah diberikan pada user melalui endpoint yang akan mengakses program backend. Data yang dikembalikan akan dicek apakah ada atau tidak, bila data ada maka program akan diteruskan ke baris 8
- Baris 8-17 : program mengisi array `tests` dan `current_test_list` dengan tes yang telah diberikan pada user
- Baris 18-28 : program mengisi array opsional dengan tes yang tidak ada pada array `current_test_list`.
- Baris 29-31 : program meminta data hasil tes yang telah diberikan pada user melalui endpoint yang akan mengakses program backend. Data yang dikembalikan akan dicek apakah ada atau tidak, bila data ada maka program akan diteruskan ke baris 32
- Baris 32-40 : program mengisi array `test_result` dengan status tes yang sudah diberikan pada user bila sudah selesai maka akan diisi `false`, namun bila belum selesai maka akan diisi `true`.
- Baris 44-48 : program mengisi data-data sesi ke dalam variabel yang berfungsi menanganinya, lalu program akan membuka modal sesi

Segmen Program 4.18 Membuat dan Memperbarui Sesi

```

01:  async createSession(){
02:      if(this.emails.length==0
03:          ||this.tests.length==0||this.start==null ||this.finish == null)
04:          Swal.fire({ title: 'Mohon Isi Semua Field!',
05:             .../ });
06:      else if(start>finish)
07:          Swal.fire({ title: 'Waktu Awal Melebihi Waktu Akhir!',
08:             .../ });
09:      else{
10:          var dateStart = new Date(this.start);
11:          var dateFinish = new Date(this.finish);
12:          var duration = (dateFinish.getTime()-
dateStart.getTime())/(1000*60)

```

```

12:         let daftarTes = []
13:         this.tests.forEach(t => { daftarTes.push(t.id) });
14:         if (this.statusAdd){
15:             axios.post(this.port+'/exam_session/create',{
16:                 "email": this.emails,
17:                 ...
18:                 "tests": daftarTes
19:             }).then((response) => {
20:                 if (response.status==200){
21:                     axios.get(this.port+'/exam_session/all')
22:                         .then(({data}) => (
23:                             this.validation(data),
24:                             Swal.fire( 'Created!', 'Sesi Baru Berhasil
Dibuat!', 'success' )
25:                                 .then(function(){
26:                                     $('#modalSession').fadeOut("fast")
27:                                     $('#bg').fadeOut("slow")
28:                                 })
29:                         ))
30:                     }else{ throw response }
31:                 }).catch( error => {
32:                     $('#spinner-modal').fadeOut("slow");
33:                     Swal.fire( 'Warning!', error.response.data,
'warning' )
34:                         });
35:                 }else{
36:                     let addTest = this.tests.filter(n
=> !this.current_test_list.includes(n))
37:                     let addingTest = []
38:                     addTest.forEach(at => { addingTest.push(at.id) });
39:                     axios.post(this.port+'/test/addTest',{
40:                         "test_id": addingTest,
41:                         "exam_session_id": this.current_exam_session
42:                     }).then((response) => {
43:                         axios.post(this.port+ '/exam_session/update',{
44:                             "updating_id": this.updating,
45:                             ...
46:                             "status": this.aktif?2:1
47:                         }).then(async (response) => {
48:                             if (response.status==200){
49:                                 for (let i = 0; i < this.test_result.length;
i++) {
50:                                     const tr = this.test_result[i];
51:                                     await axios.post(this.port+
'/test_result/update',{
52:                                         "updating_id": tr.id,
53:                                         ...
54:                                         "result": tr.result
55:                                     }).then((response) => {})
56:                                         .catch( error => { console.log('error: '
+ error) })
57:                                     }
58:                                     axios.get(this.port+ '/exam_session/all')
59:                                         .then(({data}) => (
60:                                             this.validation(data),
61:                                             Swal.fire( 'Updated!', 'Sesi Berhasil
Diperbarui!', 'success' )

```

```

62:           .then(function() {
63:             $('#modalSession').fadeOut("fast")
64:             $('#bg').fadeOut("slow")
65:           })
66:         ))
67:       } else{ throw response }
68:     }).catch( error => {});
69:   ).catch( error => {
70:     $('#spinner-modal').fadeOut("slow");
71:     Swal.fire( 'Warning!', error.response.data,
72:       'warning' );
73:     })
74:   }
75: }

```

Segmen program 4.18 berisi code untuk membuat dan mengupdate sesi. Format fungsi ini juga dipakai pada beberapa fungsi untuk membuat dan memperbarui data seperti pada master persoalan. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2-7 : program akan melakukan pengecekan apabila data yang dimasukkan valid maka program akan diteruskan ke baris 9, jika tidak maka program akan menampilkan pesan error.
- Baris 9-13 : program menginisialisasi variabel dengan isian tanggal, durasi, dan daftar tes yang akan diberikan.
- Baris 14 : program akan melakukan pengecekan apabila sedang dalam mode create maka program akan dilanjutkan pada baris 15, namun bila dalam mode update maka program akan dilanjutkan pada baris 36.
- Baris 15-20 : program memanggil endpoint untuk membuat sesi baru dengan mengirimkan data sesi baru pada program backend, lalu program akan melakukan pengecekan apakah terjadi error saat membuat sesi, bila tidak maka program dilanjutkan ke baris 21, bila error maka program akan menampilkan pesan error pada baris 30.
- Baris 21-29 : program meminta data semua sesi yang terdaftar pada program backend, variabel daftar sesi diupdate, lalu ditampilkan pesan sukses dan modal sesi ditutup.

- Baris 31-34 : apabila terjadi error selama berkomunikasi dengan program backend, maka program akan menampilkan pesan error dan menutup modal.
- Baris 36-38 : program menginisialisasi daftar tes yang ditambahkan.
- Baris 39-42 : program memanggil endpoint untuk menambahkan tes pada sesi dengan mengirimkan data tes baru pada program backend.
- Baris 43-48 : program memanggil endpoint untuk memperbarui sesi dengan mengirimkan data sesi baru pada program backend, lalu program akan melakukan pengecekan apakah terjadi error saat memperbarui sesi, bila tidak maka program dilanjutkan ke baris 49, bila error maka program akan menampilkan pesan error pada baris 67.
- Baris 49-57 : program mengupdate status semua test_result dengan memanggil endpoint ke program backend
- Baris 58-66 : program meminta data semua sesi yang terdaftar pada program backend, variabel daftar sesi diupdate, lalu ditampilkan pesan sukses dan modal sesi ditutup.
- Baris 69-72 : apabila terjadi error selama berkomunikasi dengan program backend, maka program akan menampilkan pesan error dan menutup modal.

c. Master Test

Master Test adalah fitur yang memungkinkan admin untuk melihat daftar semua test, section, dan question yang terdaftar. Admin juga dapat membuat section baru dan memperbarui data section yang lama. Selain itu, admin juga dapat membuat pertanyaan baru dengan mengupload file excel. Pertanyaan yang telah dibuat dapat diperbarui, direset, maupun dihapus satu persatu. Syarat untuk mereset pertanyaan suatu section atau menghapus pertanyaan pada section satu persatu adalah pertanyaan tersebut tidak pernah dikerjakan, pertanyaan yang sudah pernah dijawab tidak dapat dihapus. Berikut adalah potongan program yang ada pada master test.

Segmen Program 4.19 Membuat Pertanyaan

```
01: createQuestion(e) {
```

```

02:     if (this.section_id!=null){
03:         if(e.target[1].files.length<=0)
04:             Swal.fire({ title: 'Mohon Isi Semua Field!',
05:                         .../ });
06:     else{
07:         let file = e.target[1].files[0]
08:         if (file!=undefined)
09:             file.originalname = file.name
10:         let extension = file.originalname.split('.')[1]
11:         if(!['xls','xlsx'].includes(extension))
12:             Swal.fire({
13:                 title: 'Hanya dapat mengupload file dengan
ekstensi .xls atau .xlsx !',
14:                         .../ );
15:         else{
16:             let formData = new FormData()
17:             formData.append('section_id', this.section_id)
18:             formData.append('excel',file)
19:             axios.post(this.port+
'/question/createFromExcel',formData, {headers: {'Content-Type':
'multipart/form-data'}})
20:                 .then((response) => {
21:                     if (response.status==200){
22:                         let thi = this
23:                         axios.get(this.port+
'/question/all?section_id='+this.section_id)
24:                             .then(({data}) => (
25:                                 this.questionList = data,
26:                                 Swal.fire('Created!', 'Pertanyaan
Berhasil Dibuat!', 'success')
27:                                     .then(function() { thi.closeModalQuestion
() })
28:                                         ))
29:                                         }else{ throw response}
30:                                         }).catch( error => {
31:                                             $('#spinner-modal').fadeOut("slow");
32:                                             Swal.fire( 'Warning!', error.response.data,
'warning' )
33:                                         });
34:                                         }
35:                                         }
36:                                         }
37: }

```

Segmen program 4.19 berisi code untuk membuat pertanyaan. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2 : program akan melakukan pengecekan apakah sudah ada section yang dipilih, bila sudah maka program akan diteruskan ke baris 3.
- Baris 3-5 : program akan melakukan pengecekan apakah file yang akan diupload sudah dipilih, bila belum maka program akan menampilkan

pesan gagal, namun bila sudah maka program akan diteruskan ke baris 7.

- Baris 7-14 : program memasukkan file yang akan diupload ke dalam variabel dan melakukan pengecekan pada ekstensi file yang diupload. Bila yang diupload bukan file excel maka program akan menampilkan pesan error, namun bila sudah benar file excel maka program akan dilanjutkan ke baris 16.
- Baris 16-18 : program membuat variabel FormData untuk menampung isian yang akan di upload
- Baris 19-20 : program mengirimkan data file excel kepada program backend untuk diinputkan pertanyaannya kedalam database
- Baris 21 & 33 : program melakukan pengecekan apakah terjadi error pada saat menginsertkan pertanyaan di program backend, bila tidak maka program akan dilanjutkan di baris 22, namun bila error maka pada baris 29 program akan menampilkan pesan error
- Baris 22 : program menampung variable this ke dalam variabel baru untuk dipakai nantinya.
- Baris 23-24 : program meminta data semua pertanyaan dari program backend guna memperbarui data yang tampil
- Baris 25-27 : program memperbarui isi variabel daftar pertanyaan dan menampilkan pesan sukses lalu menutup modal.
- Baris 30-33 : apabila terjadi error selama berkomunikasi dengan program backend, maka program akan menampilkan pesan error dan menutup modal.

Segmen Program 4.20 Memperbarui Pertanyaan

```
01: updateQuestion() {
02:     if(this.pertanyaan=="" || this.jawaban=="" || 
03:         this.num_option=="")
04:         Swal.fire({ title: 'Mohon Isi Semua Field!', 
05:             /.../ });
06:     else{
07:         axios.post(this.port+ '/question/update', {
08:             "updating_id": this.question_id,
09:             /...
10:             "question_type": this.tipeInstruksi
```

```

10:         }).then((response) => {
11:             if (response.status==200) {
12:                 let thi = this
13:                 axios.get(this.port+
14:                   '/question/all?section_id='+this.section_id)
15:                     .then(({data}) => (
16:                         this.questionList = data,
17:                         Swal.fire('Sukses!', 'Pertanyaan Baru Berhasil
Diperbarui!', 'success')
18:                         .then(function(){ thi.closeModalUpdateQuestion()
19:                           })
20:                     )else{ throw response }
21:                 ).catch( error => {
22:                     $('#spinner-modal').fadeOut("slow");
23:                     Swal.fire( 'Warning!', error.response.data, 'warning' )
24:                 });
25:             }

```

Segmen program 4.20 berisi code untuk memperbarui pertanyaan. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2-4 : program akan melakukan pengecekan apakah data yang diinputkan sudah lengkap, bila ada data yang kosong maka program akan menampilkan pesan gagal, bila tidak maka program akan diteruskan ke baris 6.
- Baris 6-10 : program mengirimkan data pertanyaan yang ingin diperbarui kepada program backend untuk diperbarui.
- Baris 11 : program melakukan pengecekan apakah proses di program backend mengalami error, bila tidak program akan dilanjutkan pada baris 12, bila iya maka program akan dilanjutkan pada baris 19
- Baris 12 : program membuat variabel baru yang akan digunakan di dalam fungsi swal.
- Baris 13-15 : program meminta data semua pertanyaan pada persoalan yang sedang dipilih guna memperbarui data yang tampil
- Baris 16-17 : program menampilkan pesan sukses dan menutup modal.
- Baris 19 : program menampilkan pesan error.
- Baris 20-23 : apabila terjadi error selama berkomunikasi dengan program backend, maka program akan menampilkan pesan error dan menutup modal.

Segmen Program 4.21 Mereset Pertanyaan

```
01: resetQuestion() {
02:     if (this.questionList!=null) {
03:         Swal.fire({title:'Yakin mereset semua pertanyaan?',
04:             ...
05:         }).then((result) => {
06:             if (result.isConfirmed) {
07:                 $('#spinner-modal').fadeIn("slow");
08:                 axios.delete(this.port+
09:                     '/question/deletebysection/'+this.section_id)
10:                     .then((response) => {
11:                         axios.get(this.port+
12:                             '/question/all?section_id='+this.section_id)
13:                             .then(({data}) => (
14:                                 this.questionList = data,
15:                                 $('#spinner-modal').fadeOut("slow"),
16:                                 Swal.fire({ title: 'Sukses mereset
17:                     pertanyaan!',
18:                         ...
19:                     })
20:                 )
21:             }
22:         }
23:     }
24: }
```

Segmen program 4.21 berisi code untuk mereset hasil jawaban tes. Fungsi yang digunakan untuk menghapus 1 pertanyaan mirip dengan fungsi ini, perbedaannya adalah endpoint delete yang dipanggil adalah axios.delete(this.port+ '/question/deletebyid/'+question_id). Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2 : program melakukan pengecekan apakah pertanyaan dari persoalan yang sedang dipilih kosong atau tidak, bila tidak kosong maka program dilanjutkan pada baris 3
- Baris 3-6 : program menampilkan pesan konfirmasi untuk mereset pertanyaan, apabila user setuju untuk reset maka program akan dilanjutkan pada baris 7
- Baris 7 : program menampilkan animasi loading
- Baris 8-9 : program mengirimkan data persoalan yang ingin dihapus pertanyaannya kepada program backend untuk dihapus.
- Baris 10-11 : program meminta data semua pertanyaan pada persoalan yang sedang dipilih guna memperbarui data yang tampil

- Baris 12 : program memperbarui isi variabel untuk data yang ditampilkan
- Baris 13 : program menghilangkan animasi loading
- Baris 14 : program menampilkan pesan sukses

d. Logout

Logout adalah fitur yang memungkinkan semua user keluar dari akunnya. Setelah melakukan logout, maka halaman user akan dialihkan ke halaman login dan user tidak akan dapat mengakses halaman sebelum ia logout sebelum ia melakukan login kembali. Berikut adalah potongan program untuk melakukan logout.

Segmen Program 4.22 Logout

```

01: logout() {
02:     Swal.fire({ title: 'Yakin Ingin Logout?',
03:                 ... }).then(async (result) => {
04:         if (result.isConfirmed) {
05:             await axios.post("/auth/logout", { session_id:
this.$cookies.get('data_registrant').exam_session},
{ withCredentials: true });
06:             axios.defaults.headers.common['Authorization'] = '';
07:             this.$cookies.remove('refresh_token')
08:             this.$cookies.remove('data_registrant')
09:             localStorage.clear();
10:             window.location="/"
11:         }
12:     });
13: }
```

Segmen program 4.22 berisi code untuk logout/keluar dari dalam website. Fungsi logout ini digunakan pada halaman admin dan juga halaman user peserta tes. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2-4 : program menampilkan pesan konfirmasi untuk logout, apabila user setuju maka program akan dilanjutkan ke baris 5
- Baris 5 : program memanggil endpoint logout agar program backend dapat menjalankan fungsi untuk logout.
- Baris 6-10 : program menghapus semua data temporer dari user yang diset pada saat melakukan login, lalu halaman dialihkan ke halaman utama (login)

3. Halaman User

Halaman User adalah halaman yang dapat diakses oleh user yang sudah melakukan login (kecuali admin). Di dalam halaman user terdapat beberapa sub halaman yang dapat diakses oleh user peserta tes saja. Berikut akan dijelaskan beberapa aksi yang dapat dilakukan oleh user beserta potongan programnya.

a. Dashboard

Dashboard adalah halaman utama peserta tes. Halaman dashboard ini terdiri dari 2 jenis, dashboard tes dan dashboard persoalan. Dashboard tes adalah halaman dimana peserta tes dapat memilih tes yang ingin dikerjakan. Sedangkan dashboard section adalah halaman dimana peserta tes dapat memulai persoalan yang diberikan secara urut dengan menekan tombol untuk mengerjakan. Apabila peserta tes memilih tes IST ataupun EPPS, maka sebelum masuk ke halaman dashboard section, peserta tes akan diberikan form untuk mengisi data diri. Berikut adalah potongan program yang ada pada halaman dashboard.

Segmen Program 4.23 Mencari Section/Persoalan Yang Sudah Diselesaikan

```
01:  async cekSelesai(result){  
02:      if(this.id_tes==21){ this.sectionList.pop() }  
03:      if(result!=null){  
04:          result.sort((a, b) => {  
05:              let da = a.section_id, db = b.section_id;  
06:              return db - da;  
07:          });  
08:          this.now = result[0].section_id  
09:          this.hasil = []  
10:          for (let i = 0; i < this.sectionList.length ; i++) {  
11:              for (let j = 0; j < result.length; j++) {  
12:                  if(this.sectionList[i].id== result[j].section_id){  
13:                      let temp = this.sectionList[i]  
14:                      this.hasil.push({  
15:                          "section_number": temp.section_number,  
16:                          /.../  
17:                          "option_num": temp.option_num,  
18:                          })  
19:                      }  
20:                  }  
21:              }  
22:              this.hasil.sort((a, b) => {  
23:                  let da = new Date(a.createdAt), db = new  
Date(b.createdAt);  
24:                  return db - da;  
25:              });  
26:          }else{ if(this.sectionList) this.now = this.sectionList[0].id-  
1 }
```

```

27:     if(this.sectionList && this.now ==
28:         this.sectionList[ this.sectionList.length-1 ].id) {
29:             let dataReg = this.$cookies.get('data_registrant')
30:             let res = -1
31:             dataReg.test.forEach(t => { if(t[0]==this.id_tes) res =
32:                 t[1] });
33:             let data_result = null
34:             axios.get(this.port+`/test_result/${res}`)
35:             .then(({data}) => (
36:                 data_result = data,
37:                 axios.post(this.port+'/test_result/update',{
38:                     "updating_id": data_result.id,
39:                     ...
40:                     "result": data_result.result
41:                 })
42:                 .then((response) => {
43:                     axios.post(this.port+
44:                     '/exam_session/updateCurrentTest',{
45:                         "id": dataReg.exam_session,
46:                         "test_id": 0
47:                     })
48:                     .then((response) => {window.location =
49:                     '/dashboard' })
50:                 })
51:             )).catch( error => {console.log('error: '+error)})
52:         }
53:     }

```

Segmen program 4.23 berisi code untuk mengambil semua section yang sudah diselesaikan. Fungsi yang digunakan untuk mencari tes yang sudah diselesaikan mirip dengan fungsi ini, perbedaannya adalah pada fungsi mencari tes selesai tidak ada kode untuk mengupdate test_result. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2 : program akan melakukan pengecekan apabila tes sekarang merupakan tes hafalan, maka persoalan kedua akan dihapus
- Baris 3 : program akan melakukan pengecekan apakah sudah ada persoalan yang diselesaikan, bila belum ada maka program dilanjutkan ke baris 4, bila sudah maka program akan dilanjutkan ke baris 26.
- Baris 4-8 : program mengurutkan semua id persoalan yang sudah diselesaikan secara descending, lalu variabel now akan diisi dengan id persoalan terakhir yang diselesaikan.
- Baris 9-25 : program memasukkan data persoalan yang sudah diselesaikan ke dalam sebuah array, lalu diurutkan berdasarkan tanggal selesainya dari yang paling baru ke paling lama (descending)

- Baris 26 : program melakukan pengecekan apabila daftar persoalan tidak kosong, maka persoalan saat ini akan diisi dengan id persoalan pertama - 1.
- Baris 27 : program melakukan pengecekan apabila semua persoalan telah selesai dikerjakan maka program akan dilanjutkan ke baris 28.
- Baris 28-30 : program mengambil isi cookies data_registrant, lalu dari data tersebut dicari yang id tes nya sama dengan tes saat ini dan didapatkan id test_result.
- Baris 31-34 : program memanggil endpoint untuk mengambil data hasil tes dari program backend lalu memasukkannya ke variabel data_result.
- Baris 35-40 : program memanggil endpoint untuk mengupdate data hasil tes kepada program backend dengan membawa data hasil tes baru.
- Baris 41-45 : program memanggil endpoint untuk mengganti id tes yang sedang dikerjakan menjadi 0 kepada program backend, lalu user dialihkan ke halaman dashboard tes.
- Baris 47 : apabila terjadi error selama berkomunikasi dengan program backend, maka program akan menampilkan pesan error pada console.

Segmen Program 4.24 Submit Data IST

```

01: submitIstData(e) {
02:     if (this.tgl_lahir == null || this.alasan == "") {
03:         Swal.fire({ title: 'Mohon Isi Semua Field!', ...
04:     } else{
05:         $('#spinner-modal').fadeIn("slow");
06:         axios.post(this.port+'/'+ist_data/create',{
07:             "test_result_id": this.test_result_id,
08:             ...
09:             "tujuan_tes": this.alasan
10:         })
11:         .then((dataResponse) => {
12:             const thi = this
13:             Swal.fire( 'Sukses!', 'Data diri berhasil dikirimkan.', ...
14:             'success' )
15:                 .then(function(){
16:                     thi.data_ist = true,
17:                     thi.tampil = true
18:                     $('#spinner-modal').fadeOut("slow")
19:                 })
20:             }).catch( error => {console.log('error: '+error)} );
21:     }
22: }
```

Segmen program 4.24 berisi code untuk mengirimkan data diri yang akan dipakai untuk perhitungan norma. Fungsi yang digunakan untuk melakukan submit pada data diri tes EPPS mirip dengan fungsi ini, perbedaannya terletak pada data yang dikirimkan yang mana pada tes EPPS data yang dikirimkan adalah pendidikan terakhir dan jenis kelamin. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2-4 : program akan melakukan pengecekan apakah data yang diinputkan sudah lengkap, bila ada data yang kosong maka program akan menampilkan pesan gagal, bila tidak maka program akan diteruskan ke baris 8.
- Baris 6 : program menampilkan animasi loading
- Baris 7-12 : program mengirimkan data diri peserta kepada program backend untuk dimasukkan ke dalam database.
- Baris 13 : program membuat variabel baru yang akan digunakan di dalam fungsi swal.
- Baris 14 : program menampilkan pesan sukses
- Baris 15-16 : program mengisi variabel data_ist dan tampil dengan true agar dashboard dapat ditampilkan
- Baris 18 : program menghilangkan animasi loading.
- Baris 20 : apabila terjadi error selama berkomunikasi dengan program backend, maka program akan menampilkan pesan error pada console.

b. Tes

Tes adalah fitur utama dari website ini. Fitur ini memungkinkan peserta tes untuk mengerjakan semua persoalan yang ada didalamnya. Didalam tiap persoalan nantinya akan terdiri dari beberapa soal. Peserta tes dapat memilih/mengisi jawaban, pindah ke soal sebelumnya/berikutnya, dan mengirimkan hasil jawaban. Setiap kali terjadi perubahan pada jawaban user, sistem akan menyimpannya ke dalam database sehingga apabila halaman tersebut ter-refresh, soal yang sudah dijawab tidak akan ter-reset jawabannya. Berikut adalah potongan program yang ada pada halaman tes.

Segmen Program 4.25 Pindah Ke Soal Berikutnya

```
01: nextSoal(){
02:     if (this.noSoal<this.jumSoal){
03:         this.noSoal++
04:         this.jumChoice = this.pertanyaan[this.noSoal-
05:             1]["option_num"]
06:         if(this.noSoal==this.jumSoal) $('#nextBtn').text('Submit')
07:         if(this.changed) this.uploadTempAnswers();
08:         this.progress(true)
09:         if(this.pertanyaan[this.noSoal-1]['option_type']==1 &&
10:             this.pertanyaan [this.noSoal-1]['option_a'] == '-')
11:             this.$refs.textAnswer.resetText(this.jawaban[this.noSoal-1])
12:         }else{
13:             var isi = 0
14:             this.jawaban.forEach(e => { if (e != null) isi++; });
15:             if(isi!=this.jumSoal){
16:                 Swal.fire({ title: 'Ada Pertanyaan yang Belum Dijawab!',
17:                         ... }).then((result) => {
18:                             if (result.isConfirmed) {
19:                                 clearInterval(this.waktu)
20:                                 this.submitJawaban()
21:                             }
22:                         });
23:             }else{
24:                 Swal.fire({ title: 'Submit This Task?',
25:                         ... }).then((result) => {
26:                             if (result.isConfirmed) {
27:                                 clearInterval(this.waktu)
28:                                 this.submitJawaban()
29:                             }
30:                         });
31:             this.gantiPilihanJawaban()
32:         }
33:     }
34: }
```

Segmen program 4.25 berisi code untuk pindah ke soal berikutnya. Fungsi yang digunakan untuk pindah ke soal sebelumnya dan ke nomor soal yang dipilih lewat menu daftar soal mirip dengan fungsi ini, perbedaannya adalah pada kedua fungsi ada tidak ada tambahan kode untuk memanggil fungsi mengirimkan hasil jawaban. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2 : program akan melakukan pengecekan apakah soal saat ini sudah soal terakhir, bila belum maka program dilanjutkan di baris 3, namun bila sudah disoal terakhir maka program akan diteruskan ke baris 11.
- Baris 3 : program menambahkan nomor soal sebanyak 1

- Baris 4 : program mengupdate jumlah pilihan jawaban pada nomor tersebut
- Baris 5 : program melakukan pengecekan apabila sudah sampai soal terakhir, maka tombol “Selanjutnya” akan diganti dengan “Submit”
- Baris 6 : program melakukan pengecekan apabila jawaban user diganti/bertambah maka program akan memanggil fungsi untuk mengupload ke database
- Baris 7 : program memanggil fungsi untuk menambah progressbar
- Baris 8-9 : program melakukan pengecekan apabila tipe jawaban adalah teks isian maka kolom isian akan direset/diisi dengan jawaban peserta
- Baris 11-13 : program melakukan pengecekan apakah semua soal sudah dijawab, bila sudah maka program dilanjutkan ke baris 14, namun bila belum maka program dilanjutkan ke baris 22.
- Baris 14-16 : program memberikan pesan konfirmasi untuk submit jawaban, bila peserta setuju untuk submit maka program dilanjutkan ke baris 17.
- Baris 17-18 : program menghentikan timer dan memanggil fungsi submit jawaban
- Baris 22-28 : cara kerja program baris ini sama dengan pada baris 14-16, perbedaannya adalah pesan pada pesan konfirmasinya.

Segmen Program 4.26 Mengirimkan hasil jawaban

```

01: submitJawaban(){
02:     $('#spinner-modal').fadeIn("slow");
03:     for (let i = 0; i < this.jumSoal; i++) {
04:         this.jawabanFinal[i] = []
05:         this.jawabanFinal[i]["question_id"] =
this.pertanyaan[i]['id']
06:         if(this.pertanyaan[this.noSoal-1]['option_type']==1 &&
this.pertanyaan [this.noSoal-1]['option_a']=='-')
07:             this.jawabanFinal[i]["answer"] = this.jawaban[i] !=
undefined ? this.jawaban[i] : '';
08:             ...
09:         else
10:             this.jawabanFinal[i]["answer"] = this.jawaban[i]!=null ?
this.jawaban[i] .substring(0,1) : '';
11:         this.jawabanFinal[i] = Object.assign({}, this.jawabanFinal[i]);
12:     }

```

```

13:     let formData = { exam_session: this.exam_session,
14:                     section_id: this.section_id,
15:                     data: this.jawabanFinal }
16:     axios.post(this.port+'/section_result/create',{ 
17:         "test_result_id": this.test_result_id,
18:         ...
19:         "finish_date": Date.now()
20:     }).then((response) => {
21:         axios.post(this.port+
22: '/question_result/createmultiple',formData)
23:         .then((response) => {
24:             axios.post(this.port+ '/test_result/calculateresult',{
25:                 test_id: this.test_id,
26:                 email: this.email
27:             }).then((response) => {
28:                 axios.post(this.port+
29: '/section_ongoing/stopSection',{
30:                     section_id: this.section_id,
31:                     exam_session: this.exam_session
32:                 }).then((response) => {
33:                     this.$cookies.remove('current_section')
34:                     this.$cookies.remove("start_time")
35:                     $('#spinner-modal').fadeOut("slow")
36:                     Swal.fire( 'Submitted!', 'Task Successfully
Submitted.', 'success' )
37:                     .then(function(){ window.location =
38: '/section' })
39:                 })
40:             })
41:         })
42:     })
43: }).catch( error => { console.log('error: ' + error) });

```

Segmen program 4.26 berisi code untuk mengirimkan hasil jawaban peserta. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2 : program menampilkan animasi loading
- Baris 3-12 : program melakukan encode terhadap jawaban peserta kedalam format yang telah ditentukan
- Baris 13-15 : program memasukkan data id sesi, id persoalan, dan jawaban peserta ke dalam variabel yang akan dikirimkan nantinya
- Baris 16-20 : program memanggil endpoint untuk membuat hasil persoalan baru kepada program backend dengan membawa data peserta
- Baris 21-22 : program memanggil endpoint untuk memasukkan semua jawaban peserta kedalam database kepada program backend dengan membawa data yang telah disiapkan pada baris 13-15

- Baris 23-26 : program memanggil endpoint untuk menghitung hasil tes kepada program backend.
- Baris 27-30 : program memanggil endpoint untuk mengganti status persoalan yang sedang dikerjakan menjadi selesai kepada program backend.
- Baris 31-32 : program menghapus cookies yang digunakan pada pengerajan tes
- Baris 33-35 : program menghilangkan animasi loading, menampilkan pesan sukses, dan mengembalikan user ke halaman dashboard persoalan.
- Baris 39 : apabila terjadi error selama berkomunikasi dengan program backend, maka program akan menampilkan pesan error pada console.

Segmen Program 4.27 Mengupload jawaban saat ini

```

01: uploadTempAnswers(){
02:     this.jawabanTemp = "";
03:     for (let i = 0; i < this.jumSoal; i++) {
04:         this.jawabanTemp += this.jawaban[i] != null ?
this.jawaban[i] : "";
05:         this.jawabanTemp += ";";
06:     }
07:     axios.post(this.port+ '/section_ongoing/updateTempAnswers', {
08:         "section_id": this.section_id,
09:         "exam_session": this.exam_session,
10:         "temp_answers": this.jawabanTemp,
11:     }).then((response) => { this.changed = false; })
12:     .catch( error => { console.log('error: ' + error) });
13: }
```

Segmen program 4.27 berisi code untuk mengirimkan jawaban sementara ke dalam database. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2-6 : program akan merekap semua jawaban yang telah diisi dengan format jawaban1;jawaban2;jawaban3;dst.
- Baris 7-10 : program mengirimkan jawaban sementara kepada program backend untuk dimasukkan ke dalam database
- Baris 11 : program akan mengganti status jawaban diubah menjadi false. Variabel ini digunakan sebagai parameter untuk menunjukkan apakah

peserta tes menambah/mengubah jawaban, jika iya maka program akan menjalankan fungsi ini.

- Baris 12 : apabila terjadi error selama berkomunikasi dengan program backend, maka program akan menampilkan pesan error pada console.

Segmen Program 4.28 Mengambil jawaban tes saat ini

```
01: getTempAnswers() {
02:   axios.get(this.port+'/section_ongoing/getbysection/'+
03:     this.section_id +'?exam_session_id=' +this.exam_session)
04:   .then((data) => {
05:     if(data && data.data){
06:       data = data.data.filter((obj) => { return
07:         obj.start_status == 1; })[0];
08:       let temp = data.temp_answers.split(";");
09:       let temp_answers = [];
10:       for(let i = 0; i < temp.length; i++){ temp_answers[i] =
11:         temp[i] ; }
12:       this.jawaban = temp_answers
13:     }
14:   }).catch( error => { console.log('error: ' + error) });
15: }
```

Segmen program 4.28 berisi code untuk mengirimkan jawaban sementara ke dalam database. Fungsi ini akan dijalankan setiap user masuk kedalam halaman tes. Apabila sebelumnya user sudah pernah menjawab lalu halaman tersebut tidak sengaja refresh, maka jawaban akan ditarik kembali dan menyebabkan user tidak perlu menginputkannya kembali. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2-3 : program akan meminta jawaban sementara dari peserta tes kepada program backend
- Baris 4-10 : program akan memproses hasil kembalian dari program backend dan memasukkannya ke variabel jawaban.
- Baris 12 : apabila terjadi error selama berkomunikasi dengan program backend, maka program akan menampilkan pesan error pada console.

c. Biodata

Biodata adalah fitur dimana peserta tes dapat mengisi data dirinya. Apabila pada saat mengirimkan data diri ternyata masih ada yang belum diisi, maka peserta tes dapat membuka biodata dan mengisi data yang kurang. Data yang sebelumnya sudah diisi akan tampil kembali. Peserta tes juga dapat mengupload

tanda tangan digital. Berikut adalah potongan program yang ada pada halaman biodata.

Segmen Program 4.29 Inisialisasi Data Lama

```
01: initDataLama(data){  
02:     this.data_old.seni_dikuasai = data.seni_dikuasai != undefined ?  
03:         data.seni_dikuasai : {}  
04:     /.../  
05:     this.data_old.kesehatan = data.kesehatan != undefined ?  
06:         data.kesehatan : {}  
07:     this.noHp = this.data_old.no_hp  
08:     this.status_perkawinan = this.data_old.status_nikah.menikah  
09:     this.sudahMenikah = false  
10:     if (this.status_perkawinan=="sudah menikah") this.sudahMenikah =  
11:         true  
12:     /.../  
13:     this.adaKenalan = false  
14:     if(this.data_old.keterangan_kerja.kenalan_perusahaan== "Ya")  
15:         this.adaKenalan = true  
16:     if (data.pendidikan == undefined)  
17:         this.pendidikan = [{tingkat:'', jurusan:'', nama_sekolah:'',  
18:         kota:'', tahun_lulus:''}]  
19:     else  
20:         this.pendidikan = data.pendidikan  
21:     /.../  
22:     if (data.orang_terdekat == undefined)  
23:         this.orang_terdekat = [{nama:'',  
24:         alamat:'',telepon:''},{nama:'', alamat:'', telepon:''}]  
25:     else  
26:         this.orang_terdekat = data.orang_terdekat  
27:     this.ready = true  
28: }
```

Segmen program 4.29 berisi code untuk inisialisasi data dari biodata peserta tes. Fungsi ini bertujuan agar apabila peserta telah melakukan submit pada biodatanya namun ada data yang salah, maka pada saat membuka form biodata peserta tidak perlu mengisi semua dari awal. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2-7 : program menginisialisasi isi variabel dari isian textbox
- Baris 8-14 : program menginisialisasi isi variabel dari isian radio button
- Baris 15-23 : program menginisialisasi isi variabel dari isian tabel
- Baris 24 : program mengubah isi variabel ready menjadi true agar form biodata dapat ditampilkan

Segmen Program 4.30 Submit Biodata

```
01: submitForm(e) {
02:     let data = Object.fromEntries(new FormData(e.target).entries());
03:     let res = []
04:     let res2 = {}
05:     let notInclude = ["imgSign", "keadaan",
06:         "sakit_keras", ... , "kota_ttd"]
07:     Object.entries(data).map(d => {
08:         if(!d[0].includes("[") && !notInclude.includes(d[0])) {
09:             res.push([d[0], d[1]])
10:         } else if(!d[0].includes("[") && notInclude.includes(d[0])) {
11:             res2[d[0]] = d[1]
12:         }
13:     })
14:     let formData = new FormData()
15:     for (let i = 0; i < res.length; i++) {
16:         formData.append(res[i][0], res[i][1])
17:     }
18:     let kesehatan = JSON.stringify({
19:         "keadaan": res2.keadaan,
20:         "cacat": res2.cacat,
21:         "sakit_keras": res2.sakit_keras,
22:         "kapan": res2.kapan
23:     })
24:     formData.append("kesehatan", kesehatan)
25:     ...
26:     formData.append("tanda_tangan", fileTT)
27:     $('#spinner-modal').fadeIn("slow");
28:     let data_result = null
29:     axios.post(this.port+'/registrant/update', formData)
30:     .then((response) => {
31:         if (response.status==200) {
32:             axios.get(this.port+
33:             `/test_result/${this.id_tes_result}`)
34:             .then(({data}) => {
35:                 data_result = data,
36:                 axios.post(this.port+'/test_result/update',{
37:                     "updating_id": data_result.id,
38:                     ...
39:                     "result": formData
40:                 }).then((response) => {
41:                     $('#spinner-modal').fadeOut("slow");
42:                     Swal.fire('Sukses!', 'Biodata berhasil
43: disimpan.', 'success')
44:                     .then(function() { window.location =
45:                         '/dashboard' })
46:                 })
47:             } else{ throw response }
48:         }).catch( error => {
49:             $('#spinner-modal').fadeOut("slow");
50:             Swal.fire('Warning!', error.response.data, 'warning')
51:         });
52:     }
53: }
```

Segmen program 4.30 berisi code untuk mengirimkan biodata. Berikut akan dijelaskan cara kerja program urut dari baris pertama hingga baris terakhir:

- Baris 2-9 : program mengambil semua data form dan diproses terlebih dahulu untuk mendapatkan bentuk yang mudah diproses lebih lanjut

- Baris 10-18 : program membuat variabel formData untuk menampung semua data dari form untuk dikirimkan ke program backend
- Baris 19 : program menampilkan animasi loading
- Baris 20-22 : program memanggil endpoint untuk mengupdate data peserta tes dengan membawa isi biodata yang sudah diisi
- Baris 23 : program melakukan pengecekan apakah terjadi error pada saat melakukan update di program backend, bila tidak maka program dilanjutkan ke baris 24, namun bila error maka program akan menampilkan pesan error pada baris 37.
- Baris 24-26 : program memanggil endpoint untuk mengambil data hasil tes dari program backend lalu memasukkannya ke variabel data_result.
- Baris 27-31 : program memanggil endpoint untuk mengupdate data hasil tes kepada program backend dengan membawa data hasil tes baru.
- Baris 32-34 : program menghilangkan animasi loading, menampilkan pesan sukses, dan mengembalikan user ke halaman dashboard
- Baris 38-39 : apabila terjadi error selama berkomunikasi dengan program backend, maka program akan menghilangkan animasi loading dan menampilkan pesan error.

BAB V

UJI COBA DAN DEPLOYMENT

Setelah dilakukan tahapan implementasi, langkah selanjutnya adalah melakukan pengujian pada sistem, hal ini bertujuan agar hasil akhir sistem yang diimplementasikan sesuai dengan yang dibutuhkan. Setelah sistem selesai diuji coba, maka tahap berikutnya adalah deployment agar sistem dapat dijalankan secara online. Pada bab ini akan dijelaskan tentang analisis hasil uji coba dan proses deployment website tes psikologi PT. Adiputro.

5.1 Uji Coba

Pengujian adalah proses yang bertujuan untuk memastikan apakah semua fungsi sistem bekerja dengan baik dan mencari kesalahan yang mungkin terjadi pada sistem. Pengujian yang dilakukan pada website tes psikologi PT. Adiputro. menggunakan metode pengujian Black Box Testing. Black box testing atau dapat disebut juga *Behavioral Testing* adalah pengujian yang dilakukan untuk mengamati hasil input dan output dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. Pengujian ini dilakukan di akhir pembuatan perangkat lunak untuk mengetahui apakah perangkat lunak dapat berfungsi dengan baik. Berikut akan dijabarkan mengenai hasil uji coba pada beberapa fitur utama.

5.1.1 Login

Sub bab ini akan menjelaskan hasil uji coba dari fitur login yang digunakan untuk mengakses fitur-fitur utama pada website. Ketika user mengisikan email dan token tes pada field yang tersedia lalu menekan tombol login, sistem akan mencari data sesi pada database berdasarkan email dan token tes yang telah diisikan. Jika data ditemukan dan valid, sistem mengarahkan user ke halaman dashboard registrant. Berikut ini merupakan tabel uji coba pada form login.

Tabel 5. 1
Tabel Uji Coba Login

Langkah - Langkah	Ekspektasi	Realita	Hasil
--------------------------	-------------------	----------------	--------------

Mengisi field email dan token tes dengan data yang benar lalu menekan tombol Login	Halaman dialihkan ke halaman dashboard user	Halaman dialihkan ke halaman dashboard user	Pass
Mengisi pada field email dan token tes dengan data yang salah lalu menekan tombol login	Halaman tetap berada di halaman login dan diberikan pesan gagal	Halaman tetap berada di halaman login dan diberikan pesan gagal	Pass
Mengosongkan semua field lalu menekan tombol login	Halaman tetap berada di halaman login dan diberikan pesan gagal	Halaman tetap berada di halaman login dan diberikan pesan gagal	Pass

5.1.2 Rekrutmen

Sub bab ini akan menjelaskan hasil uji coba dari fitur rekrutmen yang berfungsi untuk mendaftar lowongan pekerjaan pada PT. Adiputro. Untuk melamar pekerjaan, user dapat memindai kode QR yang sudah ada pada brosur rekrutmen. Setelah memindai, user dapat memasukkan url yang diberikan dan sistem akan mengalihkan user ke halaman rekrutmen. Ketika user mengisi biodata, portofolio, dan juga lampiran lalu menekan tombol submit, maka sistem akan menyimpan semua data user ke dalam database. Semua file yang telah disertakan oleh user akan terupload pada google drive. Berikut ini merupakan tabel uji coba pada form rekrutmen.

**Tabel 5.2
Tabel Uji Coba Rekrutmen**

Langkah - Langkah	Ekspektasi	Realita	Hasil
Mengisi semua field lalu menekan tombol submit	Sistem menyimpan data pelamar, menampilkan pesan sukses, dan mengarahkan user ke halaman login	Sistem menyimpan data pelamar, menampilkan pesan sukses, dan mengarahkan user ke halaman login	Pass

5.1.3 Halaman Registrant

Sub bab ini akan menjelaskan hasil uji coba dari fitur-fitur yang ada pada halaman registrant/peserta tes yang meliputi isi biodata, kerja tes, upload kertas buram, dan layar responsif.

1. Isi biodata

Untuk mengisi biodata, user dapat menekan card isi biodata dan sistem akan mengalihkan user ke halaman pengisian biodata. User perlu mengisikan semua

data diri pada field yang tersedia, user juga dapat mengupload tanda tangan pada kolom tanda tangan. Setelah sudah yakin bahwa data diri yang diisikan sudah valid, user dapat menekan tombol Submit. Sistem akan memasukkan isi biodata yang telah diisi ke dalam database. Berikut ini merupakan tabel uji coba pada fitur isi biodata.

Tabel 5. 3
Tabel Uji Coba Isi Biodata

Langkah - Langkah	Ekspektasi	Realita	Hasil
Mengisi semua biodata lalu menekan tombol submit	Sistem menyimpan biodata, menampilkan pesan sukses, dan mengarahkan user ke halaman dashboard	Sistem menyimpan biodata, menampilkan pesan sukses, dan mengarahkan user ke halaman dashboard	Pass
Mengisi sebagian biodata lalu menekan tombol submit	Sistem menyimpan biodata, menampilkan pesan sukses, dan mengarahkan user ke halaman dashboard	Sistem menyimpan biodata, menampilkan pesan sukses, dan mengarahkan user ke halaman dashboard	Pass

2. Mengerjakan tes

Untuk mengerjakan tes, user dapat memilih card tes yang ingin dikerjakan, lalu sistem akan mengarahkan user ke halaman dashboard persoalan yang berisi semua persoalan yang ada pada tes tersebut. User dapat mengerjakan semua persoalan secara urut dengan menekan tombol “Kerjakan Persoalan Ini”, lalu sistem akan mengarahkan user ke halaman pre test yang berisi instruksi untuk mengerjakan tes. Untuk memulai tes user dapat menekan tombol mulai dan mengerjakan semua soal lalu menekan tombol submit untuk mengumpulkan hasil pekerjaan. Berikut ini merupakan tabel uji coba pada fitur mengerjakan tes.

Tabel 5. 4
Tabel Uji Coba Mengerjakan Tes

Langkah - Langkah	Ekspektasi	Realita	Hasil
Menjawab semua soal lalu menekan tombol submit	Sistem menyimpan jawaban user, menghitung hasil tes, menampilkan pesan sukses,	Sistem menyimpan jawaban user, menghitung hasil tes, menampilkan pesan sukses, dan	Pass

	dan mengarahkan user ke halaman dashboard	mengarahkan user ke halaman dashboard	
Mengosongi semua soal lalu menekan tombol submit	Sistem menyimpan jawaban user, menghitung hasil tes, menampilkan pesan sukses, dan mengarahkan user ke halaman dashboard	Sistem menyimpan jawaban user, menghitung hasil tes, menampilkan pesan sukses, dan mengarahkan user ke halaman dashboard	Pass
Menjawab sebagian soal lalu menekan tombol submit	Sistem menyimpan jawaban user, menghitung hasil tes, menampilkan pesan sukses, dan mengarahkan user ke halaman dashboard	Sistem menyimpan jawaban user, menghitung hasil tes, menampilkan pesan sukses, dan mengarahkan user ke halaman dashboard	Pass

3. Upload kertas buram

Untuk mengupload kertas buram, user dapat menekan card Upload Kertas Buram lalu sistem akan menampilkan modal untuk mengupload kertas buram. Ketika user memilih gambar dan menekan tombol upload, maka sistem akan mengupload kertas buram ke google drive. Apabila user mengupload file lagi, maka sistem akan melakukan replace terhadap file yang lama. Berikut ini merupakan tabel uji coba pada fitur upload kertas buram.

**Tabel 5.5
Tabel Uji Coba Upload Kertas Buram**

Langkah - Langkah	Ekspektasi	Realita	Hasil
Memilih 1 gambar lalu menekan tombol upload	Sistem mereplace kertas buram yang lama, menampilkan pesan sukses, dan menutup modal kertas buram	Sistem mereplace kertas buram yang lama, menampilkan pesan sukses, dan menutup modal kertas buram	Pass
Memilih 2 gambar lalu menekan tombol upload	Sistem mereplace kertas buram yang lama, menampilkan pesan sukses, dan menutup modal kertas buram	Sistem mereplace kertas buram yang lama, menampilkan pesan sukses, dan menutup modal kertas buram	Pass
Tidak memilih gambar apapun lalu menekan tombol upload	Modal tidak tertutup dan sistem menampilkan pesan gagal	Modal tidak tertutup dan sistem menampilkan pesan gagal	Pass

4. Layar Responsif

Untuk mencoba layar responsif, user dapat mengakses halaman web menggunakan berbagai macam device seperti PC, laptop, smartphone, maupun tablet. Tampilan website akan menyesuaikan ukuran layar device yang digunakan. Beberapa halaman seperti halaman login dan halaman dashboard memiliki tampilan yang berbeda antara tampilan PC/laptop dengan tampilan smartphone. Berikut ini merupakan tabel uji coba pada fitur layar responsif.

**Tabel 5.6
Tabel Uji Coba Layar Responsif**

Langkah - Langkah	Ekspektasi	Realita	Hasil
Mengakses website menggunakan PC/laptop	Tampilan website rapi	Tampilan website rapi	Pass
Mengakses website menggunakan smartphone	Tampilan website rapi	Tampilan website rapi	Pass

5.1.4 Halaman Admin

Sub bab ini akan menjelaskan hasil uji coba dari fitur-fitur yang ada pada halaman Admin yang meliputi master registrant, master tes, master exam session, dan master recruitment.

5.1.4.1 Master Registrant

Master Registrant adalah fitur admin yang berfungsi untuk mencari registrant dan melihat detail hasil tes (termasuk mencetak hasil tes ke dalam bentuk PDF). Berikut adalah hasil uji coba yang dilakukan pada master registrant.

1. Cari Registrant

Untuk melakukan pencarian registrant, admin dapat mengisikan email yang dicari ke dalam field pencarian. Admin juga dapat mengisikan beberapa huruf saja dari email yang dicari. Sistem akan menampilkan semua email yang sesuai dengan isi pada field pencarian pada tabel registrant. Berikut ini merupakan tabel uji coba pada fitur cari registrant.

Tabel 5. 7
Tabel Uji Coba Cari Registrant

Langkah - Langkah	Ekspektasi	Realita	Hasil
Mengisi email yang ada pada database ke dalam field pencarian	Tabel registrant berisi registrant yang emailnya sesuai dengan yang ada pada kolom pencarian	Tabel registrant berisi registrant yang emailnya sesuai dengan yang ada pada kolom pencarian	Pass
Mengosongkan field	Tabel registrant berisi semua registrant yang ada	Tabel registrant berisi semua registrant yang ada	Pass
Mengisi email yang tidak ada pada database ke dalam field pencarian	Tabel registrant berisi tulisan “Belum ada data tersedia”	Tabel registrant berisi tulisan “Belum ada data tersedia”	Pass

2. Print Laporan

Untuk mencetak laporan hasil tes, admin dapat mengakses halaman detail registrant dari peserta tes yang ingin dicetak hasil tesnya. Admin dapat mencetak hasil tes dengan menekan tombol Cetak PDF. Admin juga dapat mengunduh hasil tes dengan menekan tombol Unduh PDF. Sistem akan membuat PDF hasil tes lalu menampilkan halaman preview PDF apabila admin memilih tombol Cetak PDF. apabila admin memilih tombol Unduh PDF maka sistem akan mengunduh PDF ke dalam device admin. Berikut ini merupakan tabel uji coba pada fitur print laporan.

Tabel 5. 8
Tabel Uji Coba Cari Registrant

Langkah - Langkah	Ekspektasi	Realita	Hasil
Menekan tombol Cetak PDF	Sistem menampilkan halaman preview PDF	Sistem menampilkan halaman preview PDF	Pass
Menekan tombol Unduh PDF	Sistem mengunduh PDF ke dalam device admin	Sistem mengunduh PDF ke dalam device admin	Pass

5.1.4.2 Master Tes

Master Tes adalah fitur admin yang berfungsi menambah/memperbarui persoalan dan menambah/memperbarui pertanyaan untuk tiap persoalan. Berikut adalah hasil uji coba yang dilakukan pada master tes.

1. Tambah Persoalan

Untuk menambahkan persoalan baru, admin dapat menekan tombol Buat Persoalan Baru dan sistem akan membuka modal untuk membuat persoalan. Admin harus mengisikan semua data persoalan baru lalu menekan tombol Buat. Sistem akan melakukan pengecekan terhadap data yang diisi, apabila ada field yang belum diisi, sistem akan menampilkan pesan gagal, namun apabila semua field sudah diisi dengan benar maka sistem akan membuat persoalan baru ke dalam database. Berikut ini merupakan tabel uji coba pada fitur tambah persoalan.

Tabel 5. 9
Tabel Uji Coba Tambah Persoalan

Langkah - Langkah	Ekspektasi	Realita	Hasil
Mengisi semua field lalu menekan tombol Buat	Sistem menambahkan persoalan baru, menampilkan pesan sukses, dan menutup modal persoalan	Sistem menambahkan persoalan baru, menampilkan pesan sukses, dan menutup modal persoalan	Pass
Mengisi sebagian field lalu menekan tombol Buat	modal persoalan tidak tertutup dan ditampilkan pesan gagal	modal persoalan tidak tertutup dan ditampilkan pesan gagal	Pass
Mengosongkan field lalu menekan tombol Buat	modal persoalan tidak tertutup dan ditampilkan pesan gagal	modal persoalan tidak tertutup dan ditampilkan pesan gagal	Pass

2. Update Persoalan

Untuk memperbarui persoalan, admin dapat menekan tombol Update pada baris persoalan yang ingin diupdate dan sistem akan membuka modal untuk mengupdate persoalan. Admin harus mengisikan semua data persoalan yang baru lalu menekan tombol Perbarui. Sistem akan melakukan pengecekan terhadap data yang diisi, apabila ada field yang belum diisi, sistem akan menampilkan pesan gagal, namun apabila semua field sudah diisi dengan benar maka sistem akan memperbarui data persoalan pada database. Berikut ini merupakan tabel uji coba pada fitur update persoalan.

Tabel 5. 10
Tabel Uji Coba Update Persoalan

Langkah - Langkah	Ekspektasi	Realita	Hasil
Mengisi semua field lalu menekan tombol Buat	Sistem memperbarui data persoalan, menampilkan pesan sukses, dan menutup modal persoalan	Sistem memperbarui data persoalan, menampilkan pesan sukses, dan menutup modal persoalan	Pass
Mengisi sebagian field lalu menekan tombol Buat	modal persoalan tidak tertutup dan ditampilkan pesan gagal	modal persoalan tidak tertutup dan ditampilkan pesan gagal	Pass
Mengosongkan field lalu menekan tombol Buat	modal persoalan tidak tertutup dan ditampilkan pesan gagal	modal persoalan tidak tertutup dan ditampilkan pesan gagal	Pass

3. Tambah Pertanyaan

Untuk menambahkan pertanyaan baru, admin dapat menekan tombol Buat Pertanyaan Baru dan sistem akan membuka modal untuk membuat pertanyaan. Admin harus mengupload file excel dengan format yang sudah ditentukan lalu menekan tombol Buat. Sistem akan memasukkan semua persoalan yang ada pada file excel ke dalam database. Berikut ini merupakan tabel uji coba pada fitur tambah pertanyaan.

Tabel 5. 11
Tabel Uji Coba Tambah Pertanyaan

Langkah - Langkah	Ekspektasi	Realita	Hasil
Mengupload file excel dengan format yang benar lalu menekan tombol Buat	Sistem memasukkan pertanyaan baru, menampilkan pesan sukses, dan menutup modal pertanyaan	Sistem memasukkan pertanyaan baru, menampilkan pesan sukses, dan menutup modal pertanyaan	Pass
Mengupload file excel dengan format yang salah lalu menekan tombol Buat	modal pertanyaan tidak tertutup dan ditampilkan pesan gagal	modal pertanyaan tidak tertutup dan ditampilkan pesan gagal	Pass
Mengupload file selain excel dengan format yang benar lalu menekan tombol Buat	modal pertanyaan tidak tertutup dan ditampilkan pesan gagal	modal pertanyaan tidak tertutup dan ditampilkan pesan gagal	Pass

Tidak mengupload file lalu menekan tombol Buat	modal pertanyaan tidak tertutup dan ditampilkan pesan gagal	modal pertanyaan tidak tertutup dan ditampilkan pesan gagal	Pass
--	---	---	------

4. Update Pertanyaan

Untuk memperbarui pertanyaan, admin dapat menekan tombol Update pada baris pertanyaan yang ingin diupdate dan sistem akan membuka modal untuk mengupdate pertanyaan. Admin harus mengisikan semua data pertanyaan yang baru lalu menekan tombol Perbarui. Sistem akan melakukan pengecekan terhadap data yang diisi, apabila ada field yang belum diisi, sistem akan menampilkan pesan gagal, namun apabila semua field sudah diisi dengan benar maka sistem akan memperbarui data pertanyaan pada database. Berikut ini merupakan tabel uji coba pada fitur update pertanyaan.

Tabel 5. 12
Tabel Uji Coba Update Pertanyaan

Langkah - Langkah	Ekspektasi	Realita	Hasil
Mengisi semua field lalu menekan tombol Buat	Sistem memperbarui data pertanyaan, menampilkan pesan sukses, dan menutup modal pertanyaan	Sistem memperbarui data pertanyaan, menampilkan pesan sukses, dan menutup modal pertanyaan	Pass
Mengisi sebagian field lalu menekan tombol Buat	modal pertanyaan tidak tertutup dan ditampilkan pesan gagal	modal pertanyaan tidak tertutup dan ditampilkan pesan gagal	Pass
Mengosongkan field lalu menekan tombol Buat	modal pertanyaan tidak tertutup dan ditampilkan pesan gagal	modal pertanyaan tidak tertutup dan ditampilkan pesan gagal	Pass

5.1.4.3 Master Exam Session

Master Exam Session adalah fitur admin yang berfungsi mencari sesi dan menambah/memperbarui sesi. Berikut adalah hasil uji coba yang dilakukan pada master exam session.

1. Cari Sesi

Untuk melakukan pencarian sesi, admin dapat mengisikan email yang dicari ke dalam field pencarian. Admin juga dapat mengisikan beberapa huruf saja dari

email yang dicari. Sistem akan menampilkan semua email yang sesuai dengan isi pada field pencarian pada tabel sesi.. Berikut ini merupakan tabel uji coba pada fitur cari sesi.

**Tabel 5. 13
Tabel Uji Coba Cari Sesi**

Langkah - Langkah	Ekspektasi	Realita	Hasil
Mengisi email yang ada pada database ke dalam field pencarian	Tabel sesi berisi sesi yang emailnya sesuai dengan yang ada pada kolom pencarian	Tabel sesi berisi sesi yang emailnya sesuai dengan yang ada pada kolom pencarian	Pass
Mengosongkan field	Tabel sesi berisi semua sesi yang ada	Tabel sesi berisi semua sesi yang ada	Pass
Mengisi email yang tidak ada pada database ke dalam field pencarian	Tabel sesi berisi tulisan “Belum ada data tersedia”	Tabel sesi berisi tulisan “Belum ada data tersedia”	Pass

2. Tambah Sesi

Untuk menambahkan sesi baru, admin dapat menekan tombol Buat Sesi Baru dan sistem akan membuka modal untuk membuat sesi. Admin harus mengisikan semua data sesi baru termasuk tes apa saja yang akan diberikan lalu menekan tombol Buat. Sistem akan melakukan pengecekan terhadap data yang diisi, apabila ada field yang belum diisi, sistem akan menampilkan pesan gagal, namun apabila semua field sudah diisi dengan benar maka sistem akan membuat sesi baru ke dalam database, sistem juga akan mengirimkan email kepada semua email yang dibuat sesinya. Berikut ini merupakan tabel uji coba pada fitur tambah sesi.

**Tabel 5. 14
Tabel Uji Coba Tambah Sesi**

Langkah - Langkah	Ekspektasi	Realita	Hasil
Mengisi semua field lalu menekan tombol Buat	Sistem menambahkan sesi baru, menampilkan pesan sukses, dan menutup modal sesi	Sistem menambahkan sesi baru, menampilkan pesan sukses, dan menutup modal sesi	Pass

Mengisi sebagian field lalu menekan tombol Buat	modal sesi tidak tertutup dan ditampilkan pesan gagal	modal sesi tidak tertutup dan ditampilkan pesan gagal	Pass
Mengosongkan field lalu menekan tombol Buat	modal sesi tidak tertutup dan ditampilkan pesan gagal	modal sesi tidak tertutup dan ditampilkan pesan gagal	Pass

3. Update Sesi

Untuk memperbarui sesi, admin dapat menekan tombol Update pada baris sesi yang ingin diupdate dan sistem akan membuka modal untuk mengupdate sesi. Admin harus mengisikan semua data sesi yang baru termasuk menambahkan tes ataupun mengupdate status tes yang dibuka lalu menekan tombol Perbarui. Sistem akan melakukan pengecekan terhadap data yang diisi, apabila ada field yang belum diisi, sistem akan menampilkan pesan gagal, namun apabila semua field sudah diisi dengan benar maka sistem akan memperbarui data sesi pada database. Berikut ini merupakan tabel uji coba pada fitur update sesi.

Tabel 5. 15
Tabel Uji Coba Update Sesi

Langkah - Langkah	Ekspektasi	Realita	Hasil
Mengisi semua field lalu menekan tombol Buat	Sistem memperbarui data sesi, menampilkan pesan sukses, dan menutup modal sesi	Sistem memperbarui data sesi, menampilkan pesan sukses, dan menutup modal sesi	Pass
Mengisi sebagian field lalu menekan tombol Buat	modal sesi tidak tertutup dan ditampilkan pesan gagal	modal sesi tidak tertutup dan ditampilkan pesan gagal	Pass
Mengosongkan field lalu menekan tombol Buat	modal sesi tidak tertutup dan ditampilkan pesan gagal	modal sesi tidak tertutup dan ditampilkan pesan gagal	Pass

5.1.4.4 Master Recruitment

Master Recruitment adalah fitur admin yang berfungsi menambah/memperbarui lowongan kerja dan melihat detail pelamar termasuk lampiran dan portofolio yang disertakan. Berikut adalah hasil uji coba yang dilakukan pada master recruitment.

1. Tambah Lowongan Kerja

Untuk menambahkan lowongan kerja baru, admin dapat menekan tombol Buat Rekrutmen Baru dan sistem akan membuka modal untuk membuat lowongan kerja. Admin harus mengisikan semua data lowongan baru lalu menekan tombol Buat. Sistem akan melakukan pengecekan terhadap data yang diisi, apabila ada field yang belum diisi, sistem akan menampilkan pesan gagal, namun apabila semua field sudah diisi dengan benar maka sistem akan membuat rekrutmen baru ke dalam database. Berikut ini merupakan tabel uji coba pada fitur tambah lowongan kerja.

Tabel 5. 16
Tabel Uji Coba Tambah Lowongan Kerja

Langkah - Langkah	Ekspektasi	Realita	Hasil
Mengisi semua field lalu menekan tombol Buat	Sistem menambahkan lowongan kerja baru, menampilkan pesan sukses, dan menutup modal lowongan kerja	Sistem menambahkan lowongan kerja baru, menampilkan pesan sukses, dan menutup modal lowongan kerja	Pass
Mengisi sebagian field lalu menekan tombol Buat	modal lowongan kerja tidak tertutup dan ditampilkan pesan gagal	modal lowongan kerja tidak tertutup dan ditampilkan pesan gagal	Pass
Mengosongkan field lalu menekan tombol Buat	modal lowongan kerja tidak tertutup dan ditampilkan pesan gagal	modal lowongan kerja tidak tertutup dan ditampilkan pesan gagal	Pass

2. Update Lowongan Kerja

Untuk memperbarui lowongan kerja, admin dapat menekan tombol Update pada baris rekrutmen yang ingin di update dan sistem akan membuka modal untuk mengupdate lowongan kerja. Admin harus mengisikan semua data lowongan kerja yang baru lalu menekan tombol Perbarui. Sistem akan melakukan pengecekan terhadap data yang diisi, apabila ada field yang belum diisi, sistem akan menampilkan pesan gagal, namun apabila semua field sudah diisi dengan benar maka sistem akan memperbarui data lowongan kerja pada database. Berikut ini merupakan tabel uji coba pada fitur update lowongan kerja.

Tabel 5. 17
Tabel Uji Coba Update Lowongan Kerja

Langkah - Langkah	Ekspektasi	Realita	Hasil
Mengisi semua field lalu menekan tombol Buat	Sistem memperbarui data lowongan kerja, menampilkan pesan sukses, dan menutup modal lowongan kerja	Sistem memperbarui data lowongan kerja, menampilkan pesan sukses, dan menutup modal lowongan kerja	Pass
Mengisi sebagian field lalu menekan tombol Buat	modal lowongan kerja tidak tertutup dan ditampilkan pesan gagal	modal lowongan kerja tidak tertutup dan ditampilkan pesan gagal	Pass
Mengosongkan field lalu menekan tombol Buat	modal lowongan kerja tidak tertutup dan ditampilkan pesan gagal	modal lowongan kerja tidak tertutup dan ditampilkan pesan gagal	Pass

3. Detail Pelamar

Untuk melihat detail pelamar, admin dapat menekan tombol detail pada baris pelamar kerja yang diinginkan. Sistem akan mengarahkan admin ke halaman detail pelamar. Admin dapat melihat semua data pelamar yang telah dikirimkan sebelumnya. Admin juga dapat melihat file lampiran dan portofolio dengan menekan tulisan Lihat Portofolio dan Lihat Lampiran lalu sistem akan membuka file di google drive. Berikut ini merupakan tabel uji coba pada fitur detail pelamar.

Tabel 5. 18
Tabel Uji Coba Detail Pelamar

Langkah - Langkah	Ekspektasi	Realita	Hasil
Melihat portofolio pelamar dengan menekan link portofolio	Halaman dialihkan ke google drive dari link yang dilampirkan	Halaman dialihkan ke google drive dari link yang dilampirkan	Pass
Melihat lampiran pelamar dengan menekan link lampiran	Halaman dialihkan ke google drive dari file yang dilampirkan	Halaman dialihkan ke google drive dari file yang dilampirkan	Pass

5.1.5 Kendala dalam Testing

Pada saat melakukan testing terhadap website yang kami kembangkan, terdapat beberapa kesalahan yang terjadi dan mengakibatkan error pada beberapa fitur. Seiring dengan ditemukannya bug atau kesalahan pemrograman, kami juga memperbaiki kesalahan tersebut sehingga program yang kami buat dapat berjalan dengan semestinya. Kesalahan yang paling sering ditemui pada saat melakukan testing pada alat tes yang dibuat adalah

kesalahan yang bersifat human error. Kesalahan tersebut adalah kesalahan input jawaban ke dalam tes yang menyebabkan hasil tes tidak akurat dan kesalahan penilaian yang dilakukan secara manual yang menyebabkan hasil tes tidak sama. Tetapi setelah dicocokkan dengan kunci jawaban yang ada, ditemukan bahwa perhitungan tes sudah benar. Selain human error, ada juga error yang terjadi akibat compatibility browser yang digunakan. Error ini terjadi pada fungsi fitur download dan print hasil tes tertukar pada browser Mozilla Firefox. Error lain juga terjadi pada tampilan yang dibuat dimana tampilan pada browser Mozilla Firefox sedikit berbeda.

Kesalahan juga ditemukan pada kode program yang dibuat. Yang pertama adalah kesalahan perhitungan hasil tes kraepelin karena kesalahan code pada program backend. Dimana pengecekan yang dilakukan tidak sesuai dengan data yang ada sehingga menyebabkan variabel yang dikeluarkan kosong dan pada saat variabel itu digunakan, terjadilah error. Yang kedua adalah kesalahan pada fitur reset jawaban, dimana data yang direset tidak bersih dan menyebabkan data baru tertumpuk dan perhitungan hasil tes kacau. Yang ketiga adalah error pada saat menggunakan fitur download dan print hasil tes, dimana apabila tes tidak dikerjakan secara urut, maka fitur ini akan mengalami crash. Dan yang keempat adalah error pada saat memasukkan pertanyaan melalui file excel. Error ini terjadi pada saat format excel yang dimasukkan tidak sesuai dengan template format excel yang sudah ditentukan seperti kolom yang berlebih, peletakan baris pertama yang salah.

5.2 Deployment

5.2.1 Proses Deployment

Deployment adalah kegiatan yang bertujuan untuk menyebarluaskan aplikasi yang telah dikerjakan oleh para pengembang. Penyebarannya dapat melalui berbagai cara tergantung dari jenis aplikasinya. Jika aplikasi Web, maka aplikasi tersebut dihosting pada server. Jika aplikasi Mobile, akan terdapat dua deployment, deployment untuk aplikasi ke Playstore/Appstore, serta deployment API (backend) ke server. Pada pengembangan web psikologi PT. Adiputro, deployment dilakukan dengan cara menghosting program pada server.

Dalam mengembangkan web Adiputro ini, kami menggunakan Github sebagai sarana pengerjaan kolaborasi antar developer agar bisa diakses melalui internet dan melihat

perubahan-perubahan yang ada pada file code yang di-git. Kami memiliki satu Repository atau tempat menyimpan sebuah project di Github yang menyimpan file-file coding web Adiputro ini termasuk backend, backend-clock, dan frontend.

Kemudian hosting program yang dilakukan untuk website Adiputro ini menggunakan sarana VPS. VPS atau singkatan dari Virtual Private Server adalah sebuah jasa layanan cloud hosting yang disediakan oleh beberapa penyedia hosting. VPS yang digunakan ini nantinya dapat menyimpan beberapa web app atau web service. Untuk setiap web app atau web service, setiapnya disimpan dalam satu docker container. Docker container ini bertugas menyimpan dan mengeksekusi program agar bisa diakses di internet sehingga bisa diakses oleh banyak orang dimana saja.

Pada arsitektur program Adiputro yang sudah dijelaskan di bab sebelumnya, terdapat 3 folder utama yaitu backend, backend-clock, dan frontend yang mana setiap folder ini harus diserve (dinyalakan) secara terpisah agar bisa saling berkomunikasi dan program berjalan lancar. Setiap service dan app ini akan memiliki port yang berbeda-beda tetapi satu url/alamat yang sama. Yang dapat user akses hanyalah port milik frontend karena user hanya perlu mengakses websitenya.

Nantinya pada docker di VPS, folder backend, backend-clock, dan frontend akan dimasukkan ke docker container masing-masing dan ketiga container tersebut akan diserve dengan konfigurasi berbeda-beda. Untuk dapat memasukkan folder ke dalam setiap container dengan benar, kami perlu memasukkan folder coding web ke VPS lalu membuat docker container baru di ketiga folder tersebut. Cara memasukkan coding web tersebut ke VPS adalah dengan memanfaatkan Git. VPS akan mengambil file-file yang sudah dimasukkan ke repository Github dengan cara git pull ke VPS. Setelah itu setiap folder backend, backend-clock, dan frontend akan dibuatkan docker container dengan nama yang sama.

Setelah membuat docker container, di setiap container memerlukan dua file konfigurasi server yaitu Dockerfile dan docker-compose.yaml. Setiap container memiliki konfigurasi yang berbeda-beda tetapi memiliki dasar konfigurasi yang sama. Dockerfile adalah file yang menyimpan perintah-perintah yang dapat dijalankan di command line seperti melakukan copy file, memindahkan work directory, dan menjalankan command library node.js. Sedangkan file docker-compose.yaml bertugas untuk menyimpan peraturan container seperti nama container, port yang digunakan, serta url domain yang digunakan

untuk container tersebut. Kedua file ini harus ada di setiap container agar docker dapat dijalankan.

Link url atau alamat web yang digunakan pada web Adiputro ini adalah adiputropsytest.sedaap.xyz yang mana domain sedaap.xyz disewa dari website alibabacloud.com. Alibaba cloud adalah salah satu penyedia Infrastructure as a service (IaaS) yang menyediakan berbagai macam cloud computing services. Untuk web adiputro ini, menggunakan subdomain adiputro dengan domain utama sedaap.xyz. Pada akhirnya setelah konfigurasi telah dimasukkan dan setiap container telah diserve, maka pengguna dapat mengakses web ini pada link adiputropsytest.sedaap.xyz.

5.2.2 Kendala Dalam Deployment

Dalam melakukan deployment dari local ke hosting terdapat berbagai macam kendala yang dihadapi. Mulai dari permasalahan konfigurasi server yang salah hingga koneksi antar container yang tidak dapat berjalan.

Kendala pertama yang terjadi dalam melakukan deployment ke hosting ini adalah dalam kesalahan penggunaan branch pada git repository. Pada awalnya terdapat 3 branch yaitu backend, frontend dan socket-conn. Branch frontend digunakan untuk menyimpan semua versioning dari folder backend, branch frontend untuk folder frontend, dan untuk branch socket-conn untuk folder backend-clock. Yang menjadi permasalahan adalah harus ada sebuah branch yang menampung semua perubahan yang ada jadinya nantinya pada VPS dapat ditarik dari satu branch utama. Dari sana kami harus membuat sebuah branch utama gabungan dari semua branch. Dibuatlah sebuah branch baru bernama deployment yang mana ketika menggabungkan ketiga branch lama, terdapat merge conflict. Merge conflict adalah suatu kondisi dimana saat penggabungan dua file yang memiliki perubahan file yang berbeda, git tidak dapat melakukan penggabungan secara otomatis. Terdapat banyak sekali merge conflict yang terjadi saat pembuatan branch deployment sehingga terkendala beberapa hari dalam mengurus merge conflict tersebut.

Selain itu, kendala yang dihadapi saat deployment adalah kesalahan dalam mengkonfigurasi docker container yang menyebabkan alamat url adiputropsytest.sedaap.xyz tidak dapat diakses maupun docker tidak dapat diserve. Terjadi kesusahan melakukan konfigurasi file Dockerfile terutama pada container frontend. Dikarenakan frontend

menggunakan Vue.js, maka konfigurasi container juga harus mengikuti serve milik Vue.js. Beberapa kali terjadi salah memasang file Dockerfile maupun docker-compose.yaml seperti beberapa folder penting tidak terinstall atau ditemukan dan container yang tidak dapat diserve akibat router docker yang salah. Tetapi setelah mengotak-atik dan mencari solusi di internet, container ini dapat diserve dan diakses oleh user.

Kendala lain yang terjadi adalah dalam penggunaan port dan socket untuk server. Dikarenakan web adiputro ini menggunakan 3 port dalam 1 alamat, maka port yang harus diakses di server juga harus dikonfigurasi. Sering kali port yang akan dibuka di server tidak terbuka dikarenakan salah setting. Perlunya konfigurasi yang benar untuk bisa menyalakan 2 port dalam 1 container sangat penting agar bisa melakukan komunikasi antar container. Selain itu dikarenakan web ini menggunakan socket untuk koneksi real time antara frontend dengan backend untuk menerima timer server, terjadi permasalahan CORS atau kepanjangan dari Cross-Origin Resource Sharing yang mana tidak memperbolehkan koneksi antar url. Error cors ini hal sering yang harus dihadapi oleh developer web karena merupakan salah satu perlindungan website agar tidak dilakukan penyerangan yang tidak diinginkan seperti Cross Site Forgery. Setelah berkali-kali mencoba, akhirnya perlu dipasang konfigurasi cors di coding frontend dan backend sehingga bisa saling berkomunikasi di server.

BAB VI

PENUTUP

Pada bab ini akan dibahas tentang kesimpulan dan saran dari pengembangan website tes psikologi PT. Adiputro Wirasejati yang nantinya dapat dikembangkan lebih lanjut dalam perancangan berikutnya.

6.1 Kesimpulan

Setelah website tes psikologi PT. Adiputro ini selesai dikembangkan, dapat disimpulkan bahwa beberapa solusi dari permasalahan awal yang kami tawarkan telah berhasil diwujudkan, antara lain:

1. Website tes psikologi PT. Adiputro berhasil dikembangkan dengan fitur-fitur yang telah dispesifikasikan.
2. Fitur rekrutmen pada website tes psikologi PT. Adiputro berhasil dikembangkan sehingga website ini dapat merangkap sebagai website rekrutmen pegawai baru.
3. Fitur tampilan responsif pada website tes psikologi PT. Adiputro berhasil dikembangkan sehingga tes psikologi dapat dilakukan menggunakan laptop, PC, tablet, maupun smartphone.
4. Semua alat tes yang dikembangkan untuk website tes psikologi PT. Adiputro telah selesai diuji coba.
5. Website tes psikologi PT. Adiputro dapat dijalankan dengan lancar pada server milik PT. Adiputro Wirasejati.

Walaupun website tes psikologi PT. Adiputro ini masih terdapat banyak kekurangan, diharapkan dengan adanya website ini nantinya dapat dikembangkan lebih lanjut sehingga sarana tes psikologi yang telah dibangun dapat disajikan dengan lebih menarik dan mudah dipahami baik oleh peserta tes maupun oleh admin website dengan cara ditambahkan fasilitas berupa fitur yang memadai.

6.2 Saran

Dalam pengembangan website tes psikologi PT. Adiputro, pengembang menyadari masih terdapat beberapa kelemahan, dan apabila website ini dikembangkan lebih lanjut,

diharapkan dapat memperbaiki kelemahan-kelemahan yang ada. Saran yang dapat diberikan untuk mengembangkan website ini antara lain:

1. Memanfaatkan konsep templating pada tampilan dengan baik sehingga apabila terjadi suatu perubahan, developer tidak perlu mengganti disetiap halaman yang ada.
2. Menambahkan fitur tutorial untuk setiap tes yang ada sehingga peserta tes dapat memahami instruksi yang diberikan dengan baik.
3. Menambahkan fitur tampilan responsif pada halaman admin.

Saran yang diberikan ini diharapkan dapat memperbaik kinerja website tes psikologi PT. Adiputro kedepannya sehingga dapat menjadikan tes psikologi PT. Adiputro sebagai website tes psikologi yang dikenal sebagai website dengan kualitas terbaik oleh khalayak ramai.

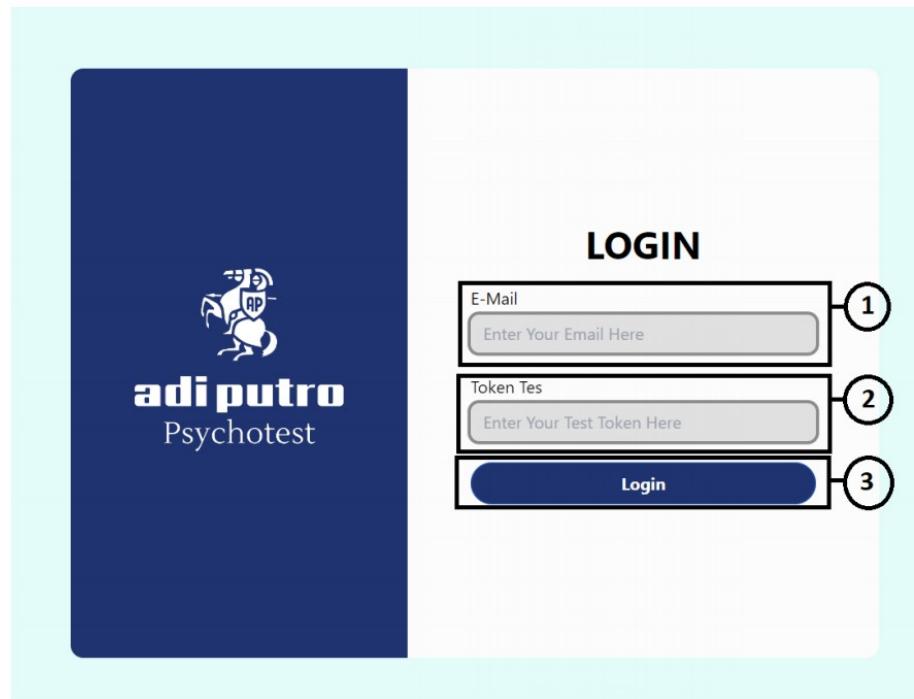
BAB VII

USER MANUAL

Bab ini berisi user manual atau panduan pengguna. Pada bab ini akan dibahas mengenai cara menggunakan program bagi pengguna. Adanya user manual diharapkan dapat mempermudah pengguna agar tidak bingung saat menggunakan program ini.

7.1 Halaman Login

Halaman login adalah halaman pertama yang akan muncul ketika website dibuka oleh seorang pengguna. Untuk dapat menggunakan fitur-fitur yang ada pada website ini, baik pengguna tes maupun admin, user wajib login ke website ini. Apabila user mencoba mengubah url untuk melakukan akses langsung terhadap page atau fitur-fitur yang ada, maka user akan dikembalikan ke halaman login ini. Berikut adalah tampilan halaman login:



**Gambar 7. 1
Form Login**

Berikut adalah langkah-langkah untuk melakukan proses login:

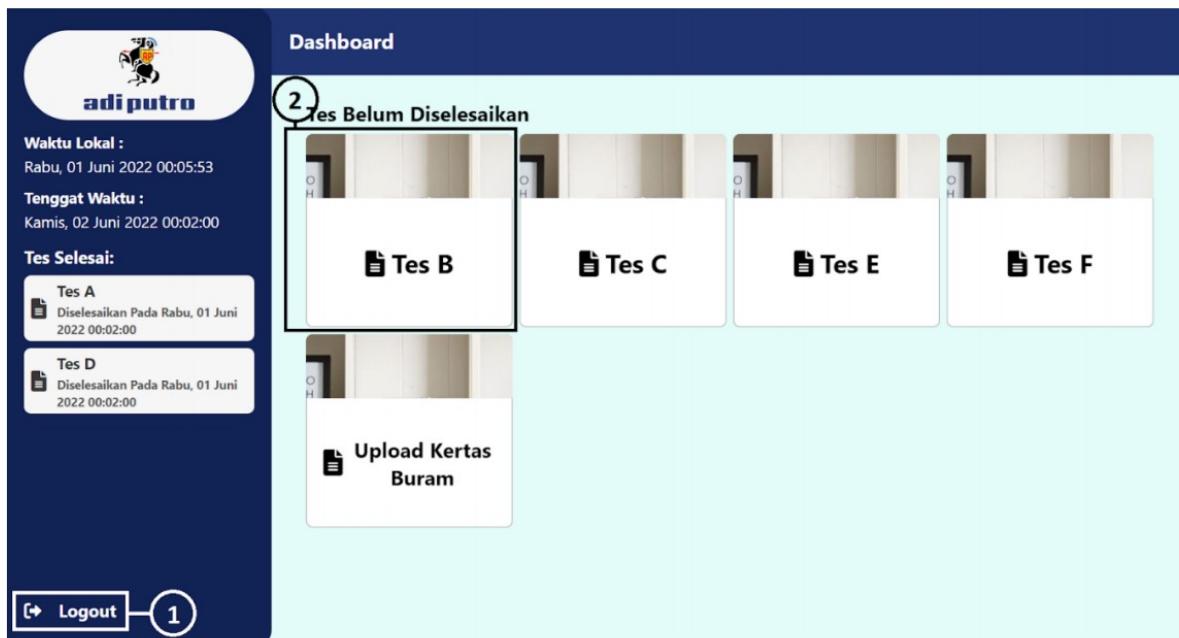
1. Isikan field email yang sudah terdaftar pada kolom email
2. Masukkan token test yang sudah dikirimkan melalui email

3. Klik tombol login untuk masuk website

Apabila email dan token sudah sesuai, user akan dialihkan ke halaman dashboard sesuai role masing-masing. Admin akan diarahkan ke dashboard admin sedangkan registrant akan diarahkan ke dashboard tes. Jika tidak sesuai maka pengguna akan menerima pesan error email atau token tidak sesuai atau error sedang tidak melakukan tes.

7.2 Halaman Dashboard Registrant

Halaman dashboard registrant ini bertujuan untuk menampilkan semua tes yang sudah dan belum dikerjakan oleh registrant. Berikut tampilan halaman dashboard registrant.



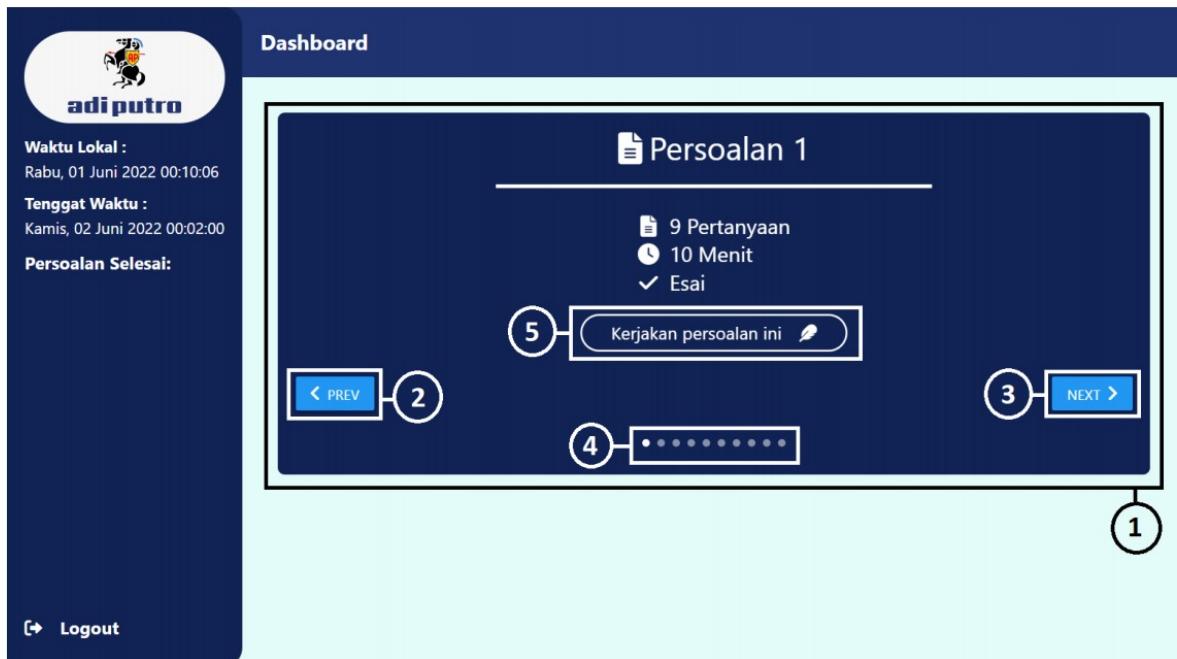
Gambar 7.2
Tampilan Halaman Dashboard Registrant

Berikut penjelasan mengenai fitur-fitur yang terdapat pada halaman dashboard.

1. Tombol Logout
2. Klik tombol Logout di paling bawah sidebar untuk keluar dari halaman website.
3. Test Card
4. Klik salah satu kotak tes untuk memilih memulai mengerjakan tes yang dipilih.

7.3 Halaman Daftar Persoalan

Halaman persoalan ini dapat diakses setelah menekan salah satu tes yang ada pada dashboard registrant. Halaman ini menampilkan semua persoalan yang harus dikerjakan registrant pada tes yang dipilih. Tampilan untuk halaman dashboard persoalan dapat dilihat pada gambar di bawah ini.



Gambar 7.3
Tampilan Halaman Daftar Persoalan

Berikut penjelasan mengenai fitur-fitur yang terdapat pada halaman ini.

1. Tampilan Persoalan
2. Bagian ini berada di dalam bagian overview milik dashboard registrant. Ditampilkan detail persoalan ada berapa pertanyaan, menit, dan merupakan tes esai atau pilihan ganda
3. Tombol Prev
4. Klik tombol ini untuk berpindah ke persoalan sebelumnya apabila ada. Apabila sekarang sedang di persoalan 1 dan ditekan tombol prev, maka akan kembali ke persoalan paling terakhir. Apabila hanya ada 1 persoalan, tombol ini tidak muncul.
5. Tombol Next

6. Klik tombol ini untuk berpindah ke persoalan selanjutnya apabila sekarang sedang di persoalan terakhir dan ditekan tombol next, maka akan kembali ke persoalan paling pertama. Apabila hanya ada 1 persoalan, tombol ini tidak muncul.
7. Navigasi Persoalan
8. Navigasi persoalan ini berupa lingkaran-lingkaran kecil yang merepresentasikan setiap persoalan yang diurutkan dari persoalan terkecil hingga terbesar. Klik setiap lingkaran untuk pindah ke persoalan yang dipilih. Apabila hanya ada 1 persoalan, navigasi ini tidak muncul.
9. Tombol Kerja Persoalan
10. Klik tombol ini untuk berpindah ke halaman pre-test untuk persiapan memulai tes untuk instruksi dan informasi lainnya. Apabila persoalan sudah selesai dikerjakan

7.4 Halaman Pre-Test

Halaman ini dapat diakses setelah menekan tombol kerjakan persoalan ini pada halaman daftar persoalan. Berikut adalah tampilan halaman pre-test.



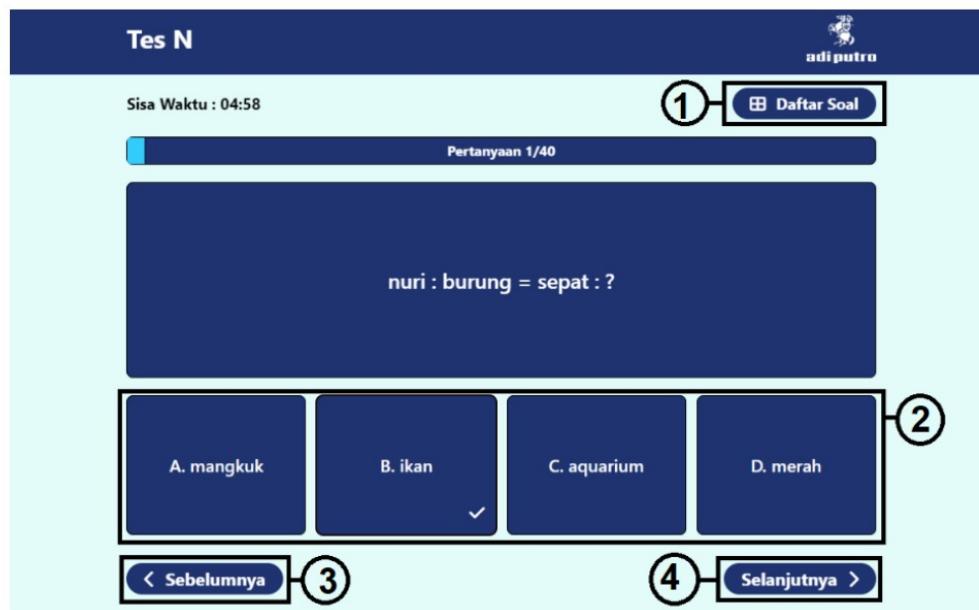
Gambar 7. 4
Tampilan Halaman Pre-Test

Berikut penjelasan mengenai fitur-fitur yang terdapat pada halaman ini.

1. Klik tombol kembali untuk kembali ke halaman daftar persoalan
2. Klik tombol mulai untuk memulai tes ini sesuai yang telah tampil informasinya

7.5 Halaman Tes Pilihan Ganda

Halaman ini dapat diakses setelah user menekan tombol mulai pada halaman pre-test. Halaman ini dapat berupa tampilan tes yang memiliki tipe tes pilihan ganda. Berikut adalah tampilan halaman tes pilihan ganda.



Gambar 7.5
Tampilan Halaman Tes Pilihan Ganda

Berikut penjelasan mengenai fitur-fitur yang terdapat pada halaman ini.

1. Klik tombol daftar soal untuk menampilkan semua daftar soal dan akan tampil nomor mana saja yang sudah dikerjakan dan masih belum dikerjakan.
2. Klik salah satu tombol pada pilihan ganda untuk memilih jawaban tes. Jawaban yang dipilih akan muncul centang putih di sebelah kanan bawah.
3. Klik tombol sebelumnya untuk berganti ke soal sebelumnya. Jika sekarang adalah pertanyaan pertama, maka tombol ini tidak akan membawa kemana-mana
4. Klik tombol selanjutnya untuk berganti ke soal setelahnya. Jika sekarang adalah pertanyaan terakhir, maka tombol ini tidak akan membawa kemana-mana

7.6 Tampilan Daftar Soal

Halaman ini dapat diakses setelah user menekan tombol mulai pada halaman pre-test. Halaman ini dapat berupa tampilan tes yang memiliki tipe tes pilihan ganda. Berikut adalah tampilan halaman daftar soal.

Daftar Soal				
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	30
31	32	33	34	35

Gambar 7.6

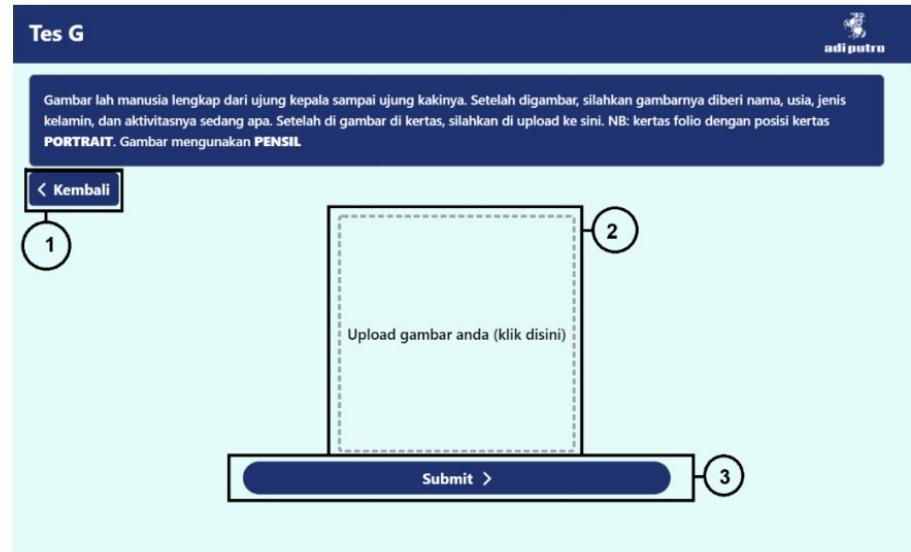
Tampilan Daftar Soal

Berikut penjelasan mengenai fitur-fitur yang terdapat pada komponen ini.

1. Klik salah satu kotak yang memiliki angka untuk berpindah ke persoalan yang dipilih. Warna putih menandakan soal belum dijawab, warna biru menandakan soal sudah dijawab, dan warna kuning menandakan sekarang berada pada soal tersebut.

7.7 Halaman Tes gambar

Halaman ini dapat diakses setelah user menekan tombol mulai pada halaman pre-test. Halaman ini dapat berupa tampilan tes khusus yang ditujukan agar user dapat menggambar di kertas lalu difoto kemudian foto tersebut dimasukkan ke lembar jawaban ini. Berikut adalah tampilan halaman tes gambar.



Gambar 7.7
Tampilan Tes Gambar

Berikut penjelasan mengenai fitur-fitur yang terdapat pada halaman ini.

1. Klik tombol kembali untuk kembali ke halaman pre-test.
2. Klik area upload gambar untuk menambahkan jawaban gambar. File yang diperbolehkan upload hanyalah file gambar saja.
3. Klik tombol submit untuk menyelesaikan persoalan ini dan gambar akan disimpan program.

7.8 Halaman Form Biodata

Halaman ini dapat diakses saat user menekan Test Card Biodata. Halaman ini berisi banyak inputan yang berkaitan dengan biodata penggerja tes. Berikut adalah tampilan halaman form biodata.

The screenshot shows a 'Biodata' form interface. At the top left is a 'Kembali' button (1). At the top right is a logo for 'adiputro' (2). The main form area contains 14 fields for personal information:

- 1. NAMA LENGKAP : Nama Lengkap
- 2. NAMA PANGGILAN : Nama Panggilan
- 3. ALAMAT DI KTP : Alamat di KTP
- 4. ALAMAT SEKARANG : Alamat Sekarang
- 5. TELEPON/HP : 081234567890
- 6. TEMPAT & TGL LAHIR : Mataram, 01/06/2000
- 9. AGAMA : Agama
- 10. USIA : 20
- 11. TINGGI BADAN : 150
- 12. BERAT BADAN : 50
- 13. NO KTP / SIM / PASPOR : 0123456789
- 14. GOLONGAN DARAH : A

Below the main form, there are three additional input fields for current address:

- Nama : Nama Lengkap
- Alamat : Alamat
- No. Telepon : 081234567890

A statement at the bottom reads: "Bawa segala biodata yang saya isi di atas adalah benar adanya."

At the bottom right are five numbered buttons:

3. A rectangular button labeled "Kota , 01 Juni 2022"
4. A large rectangular button labeled "Pelamar," with a placeholder for a signature and a "Unggah Tanda Tangan" button below it.
5. A rectangular button labeled "Kirim"

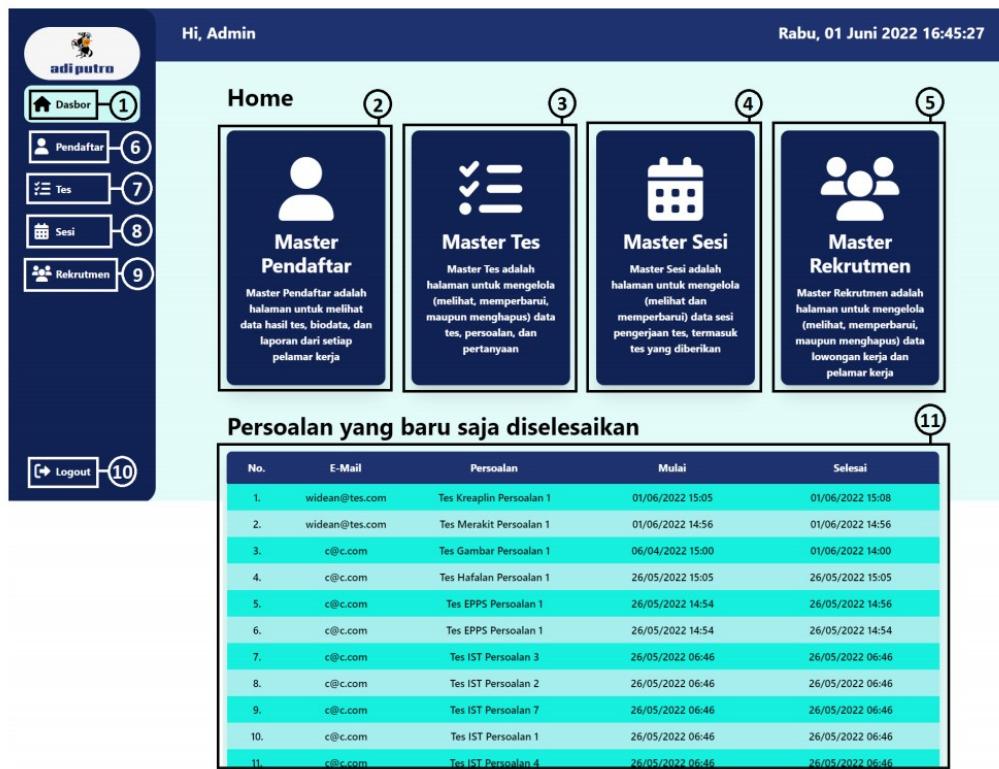
Gambar 7.8
Tampilan Halaman Form Biodata

Berikut penjelasan mengenai fitur-fitur yang terdapat pada halaman ini.

1. Klik tombol kembali untuk kembali ke halaman dashboard
2. Isi setiap field yang kosong dengan data diri yang sesuai.
3. Isi field kota dengan tempat penggerjaan tes sekarang.
4. Klik tombol unggah tanda tangan untuk menambahkan gambar tanda tangan
5. Klik tombol kirim untuk menyelesaikan form biodata.

7.9 Halaman Dashboard Admin

Halaman ini dapat diakses setelah login apabila user yang login adalah seorang admin. Halaman ini adalah halaman utama dimana admin dapat melakukan navigasi banyak master user. Berikut adalah tampilan halaman dashboard admin.



Gambar 7.9
Tampilan Halaman Dashboard Admin

Berikut penjelasan mengenai fitur-fitur yang terdapat pada halaman ini.

1. Klik tombol dasbor untuk pergi ke halaman dashboard admin
2. Klik card Master Pendaftar untuk masuk ke halaman Master Pendaftar
3. Klik card Master Tes untuk masuk ke halaman Master Tes
4. Klik card Master Sesi untuk masuk ke halaman Master Sesi
5. Klik card Master Rekrutmen untuk masuk ke halaman Master Rekrutmen
6. Klik navigasi Master Pendaftar untuk masuk ke halaman Master Pendaftar
7. Klik navigasi Master Tes untuk masuk ke halaman Master Tes
8. Klik navigasi Master Sesi untuk masuk ke halaman Master Sesi
9. Klik navigasi Master Rekrutmen untuk masuk ke halaman Master Rekrutmen

10. Klik tombol logout untuk keluar dari website dan masuk ke halaman login.
11. Tabel persoalan yang baru saja diselesaikan

7.10 Halaman Master Registrant

Halaman ini dapat diakses setelah admin menekan tombol Master Registrant pada halaman dashboard atau side navigation bar. Halaman ini bertujuan untuk melakukan proses Create, Update, dan Delete data Registrant. Berikut adalah tampilan halaman master registrant.

The screenshot shows a web application interface for managing registrants. At the top, there is a header with 'Hi, Admin' on the left and the date 'Jumat, 20 Mei 2022 20:47:44' on the right. On the far left is a vertical sidebar with icons for Home, Settings, Calendar, and Groups. The main content area has a title 'Registrant'. It features a search bar labeled 'Cari Nama Registrant:' with a placeholder 'Cari registrant disini...'. Below the search bar is a table with six rows of registrant data. The columns are 'E-Mail' (containing email addresses like 'a@a.com', 'a@tes.com', etc.), 'Status' (all listed as 'Aktif'), and 'Aksi' (each row contains a blue button labeled 'Info'). Three numbered circles point to specific elements: circle 1 points to the search bar, circle 2 points to the table, and circle 3 points to one of the 'Info' buttons.

E-Mail	Status	Aksi
a@a.com	Aktif	Info
a@tes.com	Aktif	Info
admin@admin.com	Aktif	Info
b@b.com	Aktif	Info
b@tes.com	Aktif	Info
c13180041@john.petra.ac.id	Aktif	Info

Gambar 7. 10
Tampilan Halaman Master Registrant

Berikut penjelasan mengenai fitur-fitur yang terdapat pada halaman ini.

1. Isi kolom cari nama registrant untuk melakukan pencarian registrant berdasarkan nama.
2. Tabel semua registrant yang ada pada database.
3. Klik tombol info di sebuah kolom registrant untuk menampilkan detail milik registrant tersebut.

7.11 Halaman Detail Registrant

Halaman ini dapat diakses setelah admin menekan tombol info pada tabel Master Registrant. Halaman ini menampilkan info-info penting serta laporan pribadi registrant. Berikut adalah tampilan halaman detail registrant.

The screenshot displays the 'Registrant Detail' page with the following numbered annotations:

1. Lihat Biodata button.
2. Nama Tes dropdown menu showing 'Tes IST'.
3. Hasil Tes section showing test details: E-Mail: a@tes.com, Tanggal Mulai: 13/05/2022 19:17, Nama: abc, Tanggal Selesai: 13/05/2022 22:02, Jenis Tes: Tes IST.
4. Test results table:

No. Seksi	Tipe Jawaban	Mulai	Selesai	Skor	Aksi
1	Pilihan Ganda	13/05/2022 21:31	13/05/2022 21:31	9/20	! Reset
2	Pilihan Ganda	13/05/2022 21:32	13/05/2022 21:33	10/20	! Reset
3	Pilihan Ganda	13/05/2022 21:33	13/05/2022 21:33	8/20	! Reset
4	Esai	13/05/2022 21:33	13/05/2022 21:34	15/16	! Reset
5	Esai	13/05/2022 21:34	13/05/2022 21:35	6/20	! Reset
5. Reset button for the first test result row.
6. Cetak PDF button.
7. Unduh PDF button.
- Laporan section showing:
 - PT. Adiputro Wirasejati logo and address: Jl. Raya Balearjosari no. 35, Malang, Jawa Timur - Indonesia 65126 +62 (0) 341 491 139 | Adiputrogroup.com
 - Hasil Tes IST table:

Aspek	RW	SW	Kategori
SE	9	92	Rendah
WA	10	99	Sedang
AN	8	95	Sedang
GE	15	98	Sedang
ME	10	100	Cukup
RA	6	90	Rendah
ZR	11	104	Cukup
FA	7	93	Rendah
WU	12	109	Tinggi
 - Hasil Tes IST table (repeated):

Aspek	RW	SW	Kategori
SE	9	92	Rendah
WA	10	99	Sedang
AN	8	95	Sedang
GE	15	98	Sedang
ME	10	100	Cukup
RA	6	90	Rendah
ZR	11	104	Cukup
FA	7	93	Rendah
WU	12	109	Tinggi
 - Grafik IST line chart showing scores over time.

Gambar 7.11
Tampilan Halaman Detail Registrant

Berikut penjelasan mengenai fitur-fitur yang terdapat pada halaman ini.

1. Klik tombol lihat biodata untuk menampilkan halaman hasil biodata registrant.
2. Klik drop down nama tes lalu pilih nama tes yang ingin dipilih. Drop down ini digunakan untuk menampilkan hasil tes yang dipilih.
3. Tampilan hasil tes yang dipilih
4. Tabel hasil tiap persoalan dari tes
5. Klik tombol reset tiap persoalan untuk mengulang persoalan yang diinginkan.
6. Klik tombol cetak pdf untuk membuka form print untuk bisa print hasil tes.
7. Klik tombol unduh pdf untuk mengunduh hasil tes dalam bentuk pdf.

7.12 Halaman Hasil Biodata

Halaman ini dapat diakses setelah menekan tombol Lihat Biodata pada halaman Detail Registrant. Halaman ini menampilkan biodata yang sudah diisi oleh registrant. Berikut adalah tampilan halaman hasil biodata.

1. NAMA LENGKAP	: Lawrence Patrick Santo	9. AGAMA	: kristen
2. NAMA PANGGILAN	: Lawrence	10. USIA	: 20 Tahun
3. ALAMAT DI KTP	: Jalan Jalan II	11. TINGGI BADAN	: 164 cm
4. ALAMAT SEKARANG	: Jalan Jalan III	12. BERAT BADAN	: 50.2 kg
5. TELEPON/HP	: 0812345678	13. NO KTP / SIM / PASPOR	: 981231515214561
6. TEMPAT & TGL LAHIR	: Surabaya, 01 Januari 1970	14. GOLONGAN DARAH	: O
7. KEWARGANEGARAAN	: WNI	15. NO NPWP	: 123456789
8. SUKU	: tionghoa	16. E-MAIL	: abc@def.com
17. Keadaan Kesehatan pada saat ini : baik/kurang sesuai Apakah anda pernah sakit koras/kecelakaan? ya/tidak, -/januari 2022			
18. Apakah anda belum menikah/menikah? ya/tidak Nama Suami/Istri : perempuan laki Nama Anak : anak saya			

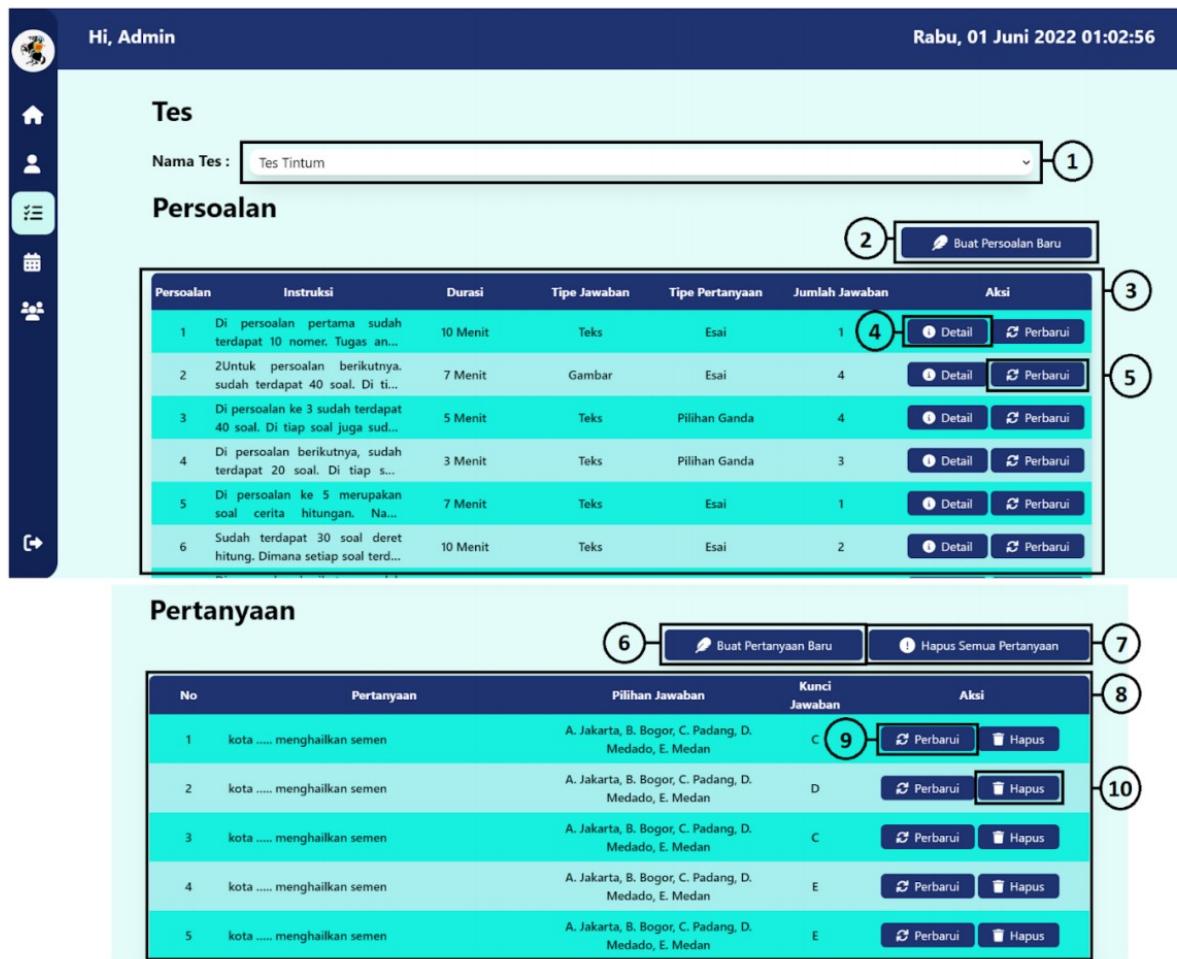
Gambar 7.12
Tampilan Halaman Hasil Biodata

Berikut penjelasan mengenai fitur-fitur yang terdapat pada halaman ini.

1. Tampilan biodata milik registrant
2. Klik tombol kembali untuk kembali ke halaman

7.13 Halaman Master Tes

Halaman ini dapat diakses setelah admin menekan tombol Master Test pada halaman dashboard atau side navigation bar. Halaman ini bertujuan untuk melakukan proses Create, Update, dan Delete data Test, Section, dan Question. Berikut adalah tampilan halaman master tes.



Hi, Admin Rabu, 01 Juni 2022 01:02:56

Tes

Nama Tes : Tes Tintum 1

Persoalan

Persoalan	Instruksi	Durasi	Tipe Jawaban	Tipe Pertanyaan	Jumlah Jawaban	Aksi
1	Di persoalan pertama sudah terdapat 10 nomer. Tugas an...	10 Menit	Teks	Esai	1	4 Detail 5 Perbarui
2	Untuk persoalan berikutnya, sudah terdapat 40 soal. Di ti...	7 Menit	Gambar	Esai	4	4 Detail 5 Perbarui
3	Di persoalan ke 3 sudah terdapat 40 soal. Di tiap juga s...	5 Menit	Teks	Pilihan Ganda	4	4 Detail 5 Perbarui
4	Di persoalan berikutnya, sudah terdapat 20 soal. Di tiap s...	3 Menit	Teks	Pilihan Ganda	3	4 Detail 5 Perbarui
5	Di persoalan ke 5 merupakan soal cerita hitungan. Na...	7 Menit	Teks	Esai	1	4 Detail 5 Perbarui
6	Sudah terdapat 30 soal deret hitung. Dimana setiap soal terd...	10 Menit	Teks	Esai	2	4 Detail 5 Perbarui

Pertanyaan

No	Pertanyaan	Pilihan Jawaban	Kunci Jawaban	Aksi
1	kota menghailkan semen	A. Jakarta, B. Bogor, C. Padang, D. Medado, E. Medan	C	6 Perbarui 7 Hapus
2	kota menghailkan semen	A. Jakarta, B. Bogor, C. Padang, D. Medado, E. Medan	D	6 Perbarui 7 Hapus
3	kota menghailkan semen	A. Jakarta, B. Bogor, C. Padang, D. Medado, E. Medan	C	6 Perbarui 7 Hapus
4	kota menghailkan semen	A. Jakarta, B. Bogor, C. Padang, D. Medado, E. Medan	E	6 Perbarui 7 Hapus
5	kota menghailkan semen	A. Jakarta, B. Bogor, C. Padang, D. Medado, E. Medan	E	6 Perbarui 7 Hapus

Gambar 7. 13
Tampilan Halaman Master Tes

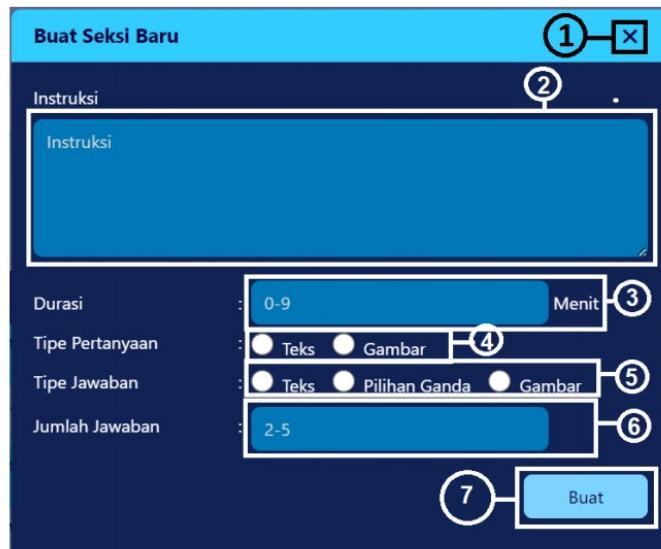
Berikut penjelasan mengenai fitur-fitur yang terdapat pada halaman ini.

1. Klik dropdown kolom nama test untuk melakukan pencarian Section dan Question berdasarkan nama tes. Section di sini adalah semua persoalan yang ada pada tes yang dipilih,
2. Klik tombol buat seksi (section) baru untuk membuat section baru untuk tes yang dipilih.

3. Tabel data semua section dari tes yang dipilih.
4. Klik tombol detail untuk melihat data yang dimiliki section yang dipilih.
5. Klik tombol perbarui untuk mengganti data section yang dipilih.
6. Klik tombol buat pertanyaan baru baru untuk menambah pertanyaan baru pada section yang dipilih. Untuk menambah pertanyaan bisa menggunakan file excel yang memiliki format input soal.
7. Klik tombol hapus semua pertanyaan untuk menghapus semua pertanyaan yang ada pada section yang dipilih.
8. Tabel data semua pertanyaan di section tersebut.
9. Klik tombol perbarui untuk mengganti data pertanyaan yang dipilih.
10. Klik tombol hapus untuk menghapus pertanyaan yang dipilih.

7.14 Komponen Buat Seksi Baru

Komponen ini akan muncul ketika tombol buat seksi baru pada halaman Master Tes ditekan. Berikut adalah tampilan komponen buat seksi baru.



Gambar 7. 14
Tampilan Komponen Buat Seksi Baru

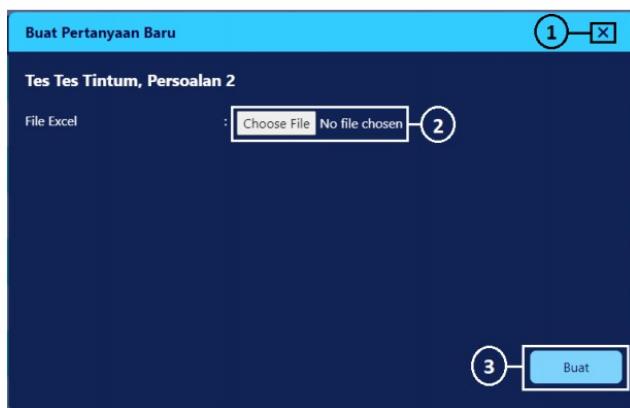
Berikut penjelasan mengenai fitur-fitur yang terdapat pada komponen ini.

1. Klik tombol silang untuk menutup komponen ini.
2. Isi instruksi dari seksi yang ingin ditambah.

3. Isi durasi dalam menit dari seksi yang ingin ditambah.
4. Pilih tipe pertanyaan dengan klik pada teks atau gambar.
5. Pilih tipe jawaban dengan klik pada teks atau pilihan ganda atau gambar.
6. Isi jumlah jawaban dengan angka berapa banyak pilihan jawaban yang ada.
7. Klik tombol buat untuk membuat seksi baru dari tes yang sedang dipilih sekarang.

7.15 Komponen Pertanyaan Baru

Komponen ini akan muncul ketika tombol buat pertanyaan baru pada halaman Master Tes ditekan. Berikut adalah tampilan komponen buat pertanyaan baru.



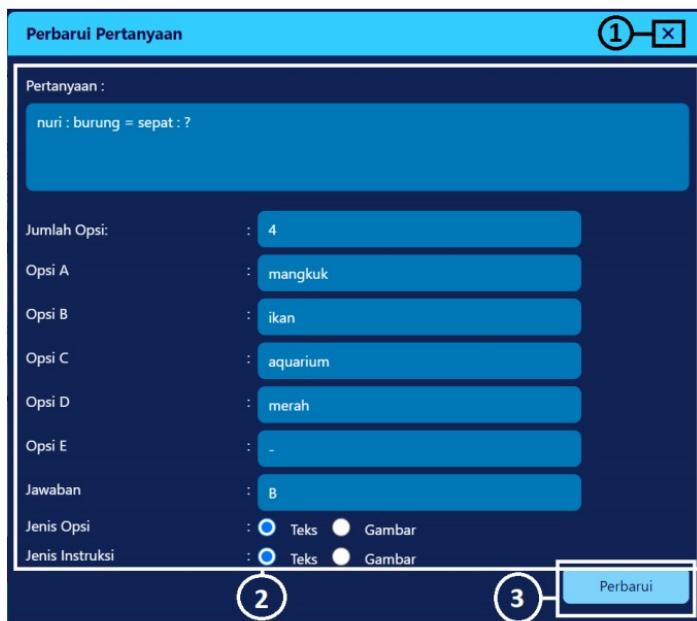
Gambar 7. 15
Tampilan Komponen Pertanyaan Baru

Berikut penjelasan mengenai fitur-fitur yang terdapat pada komponen ini.

1. Klik tombol silang untuk menutup komponen ini.
2. Klik tombol choose file untuk mengunggah file excel yang berisi pertanyaan yang akan diinsertkan ke section yang sudah dipilih.
3. Klik tombol buat untuk membuat semua pertanyaan baru dari excel untuk section yang sudah dipilih.

7.16 Komponen Perbarui Pertanyaan

Komponen ini akan muncul ketika tombol perbarui pertanyaan pada halaman Master Tes ditekan. Berikut adalah tampilan komponen perbarui pertanyaan.



Gambar 7. 16
Tampilan Komponen Perbarui Pertanyaan

Berikut penjelasan mengenai fitur-fitur yang terdapat pada komponen ini.

1. Klik tombol silang untuk menutup komponen ini.
2. Ganti data pertanyaan yang ingin diperbarui. Akan tampil data yang sebelumnya pada setiap field.
3. Klik tombol perbarui untuk memperbarui pertanyaan yang dipilih.

7.17 Halaman Master Sesi

Halaman ini dapat diakses setelah admin menekan tombol Master Sesi pada halaman dashboard atau side navigation bar. Halaman ini bertujuan untuk melakukan proses Create, Update, dan Delete data Exam Session dari Pengerja Tes. Berikut adalah tampilan halaman master registrant.

E-Mail	Mulai	Berakhir	Durasi	Token	Status	Aksi
widean2@yes.com	12/05/2022 19:21	13/05/2022 19:21	24 Jam	nRZNGnwmj89AkchMDiGow8	Aktif	Perbarui
a@tes.com	13/05/2022 19:17	25/05/2022 08:10	333.3 Jam	2x79tY5kBV7tS1PM4Vuq9a	Aktif	Perbarui
b@tes.com	13/05/2022 19:17	25/05/2022 08:10	333.3 Jam	vdtzMyTMh4BTiQ7h61uT3s	Aktif	Perbarui
c@tes.com	13/05/2022 19:17	25/05/2022 08:10	333.3 Jam	6BFKeB448SgjanEWwwjCRb	Aktif	Perbarui
feybiola@tes.com	13/05/2022 19:17	25/05/2022 08:10	333.3 Jam	nz6VmQILnPH2qogWb2CGwo	Aktif	Perbarui
priskila@tes.com	13/05/2022 19:17	25/05/2022 08:10	333.3 Jam	7fft3xk7iCwqap1mLXEH3m	Aktif	Perbarui

Gambar 7. 17
Tampilan Halaman Master Sesi

Berikut penjelasan mengenai fitur-fitur yang terdapat pada halaman ini.

1. Isi kolom cari nama email untuk melakukan pencarian exam session berdasarkan email.
2. Klik tombol buat sesi baru untuk membuat exam session baru.
3. Tabel data semua exam session yang masih aktif maupun tidak.
4. Klik tombol perbarui untuk mengganti data exam session yang diinginkan.

7.18 Komponen Sesi Baru dan Perbarui Sesi

Komponen ini dapat diakses setelah admin menekan tombol buat sesi baru pada halaman Master Sesi untuk membuka komponen buat sesi baru. Sedangkan untuk komponen perbarui sesi, admin harus menekan tombol perbarui sesi di tabel sesi pada halaman Master Sesi. Berikut adalah tampilan komponen buat sesi baru dan perbarui sesi.

A

Buat Sesi Baru

Email : email@tes.com X

Tanggal Tes
Dari : hh/bb/tttt -- Sampai : hh/bb/tttt --

Paket Tes : Paket Tes Opt, Paket S1, Paket Teknik

Tes Opsiional : Tes Berhitung, Tes EPPS X, Tes SDI X, Tes MMPI X, Tes Kreaplin X, Tes Biodata X, Tes Gambar X, Tes Hafalan X

Daftar Tes : Tes Tintum X, Tes EPPS X, Tes SDI X, Tes MMPI X, Tes Kreaplin X, Tes Biodata X, Tes Gambar X, Tes Hafalan X

Buat

B

Perbarui Sesi

Email : a@tes.com

Tanggal Tes
Dari : 13/05/2022 12.17 Sampai : 25/05/2022 01.10

Tes Opsiional : Tes EPPS X, Tes IST, Tes Tintum X

Daftar Tes : Tes Tintum X

Status Aktif : 2

Perbarui

Gambar 7.18
(a) Komponen Sesi Baru (b) Komponen Perbarui Sesi

Versi pertama dari komponen ini adalah tampilan untuk membuat sesi baru (gambar 7.18 (a)). Berikut penjelasan mengenai fitur-fitur yang terdapat pada komponen Sesi Baru.

1. Klik tombol silang untuk menutup komponen ini.
2. Tampilan email apa saja yang akan dibuat. Klik tombol silang untuk menghapus email dari list.
3. Isi email valid yang digunakan registrant untuk mengerjakan tes.
4. Tekan tombol tambahkan untuk menambahkan email ke fitur nomor 1.
5. Isi kolom tanggal awal dan akhir tes.
6. Klik salah satu paket tes yang sudah disediakan program. Ketika klik satu program paket tes, akan ditambahkan beberapa tes ke daftar tes (fitur nomor 8).
7. Klik dropdown tes opsional lalu pilih tes yang ingin ditambahkan. Kemudian tekan tombol tambahkan. Tes yang dipilih akan bertambah ke daftar tes (fitur nomor 8)
8. Daftar tes yang ada. Klik tombol silang untuk menghilangkan tes dari daftar tes.
9. Klik tombol buat untuk membuat sesi tes baru. Setiap sesi akan dikirimkan token tes melalui email masing-masing. Setiap sesi ini akan mengerjakan semua tes yang ada di daftar tes dan dapat mengerjakannya dari kurun waktu awal dan akhir.

Versi kedua dari komponen ini adalah tampilan untuk memperbarui sesi (gambar 7.18 (b)).

Berikut penjelasan mengenai fitur-fitur yang terdapat pada komponen Perbarui Sesi.

1. Klik tombol silang pada tes yang dipilih untuk menghapus dari daftar tes

2. Tekan pada tombol status aktif untuk menonaktifkan ataupun mengaktifkan sesi. Apabila tombol berwarna biru maka sesi aktif dan berlaku kebalikannya.
3. Tekan tombol perbarui untuk memperbarui data sesi.

7.19 Halaman Master Rekrutmen

Halaman ini dapat diakses setelah admin menekan tombol Master Rekrutmen pada halaman dashboard atau side navigation bar. Halaman ini bertujuan untuk melakukan proses Create, Update, dan Delete data Rekrutmen dan lamaran pekerjaan. Berikut adalah tampilan halaman master rekrutmen.

QR Barcode	Nama	Lowongan	Dibuat Pada	Status	Aksi
	Lamaran ini bagus	Manager,Kasir	18/03/2022 02:03	Open	Detail Perbarui
	Lamaran ini bagus	Manager,Kasir	18/03/2022 02:03	Open	Detail Perbarui

Email	Nama	Jenis Kelamin	Melamar Pada	Aksi
aa@gmail.com	Lawrence Patrick	Laki-Laki	17/03/2022 22:23	Detail
aaa@gmail.com	Lawrence Patrick	Laki-Laki	20/03/2022 10:55	Detail
aaaa@gmail.com	Lawrence Patrick	Laki-Laki	20/03/2022 11:37	Detail

Gambar 7. 19
Tampilan Halaman Master Rekrutmen

Berikut penjelasan mengenai fitur-fitur yang terdapat pada halaman ini.

1. Klik tombol buat rekrutmen baru untuk membuat rekrutmen pekerjaan baru.
2. Tabel data rekrutmen pekerjaan yang masih aktif atau tidak aktif.
3. Klik tombol detail untuk melihat semua data yang ada pada lamaran yang dipilih.
4. Klik tombol perbarui untuk mengubah data yang ingin diganti pada lamaran pekerjaan seperti mengganti tanggal awal atau tanggal akhir.
5. Tabel data pelamar kerja yang sudah melakukan submit form rekrutmen.

6. Klik tombol detail untuk melihat data pelamar kerja.

7.20 Komponen Rekrutmen Baru

Komponen ini dapat diakses setelah admin menekan tombol buat rekrutmen baru pada halaman Master Rekrutmen. Komponen ini digunakan untuk membuat lowongan/rekrutmen pekerja. Berikut adalah tampilan komponen rekrutmen baru.



Gambar 7. 20
Tampilan Komponen Rekrutmen Baru

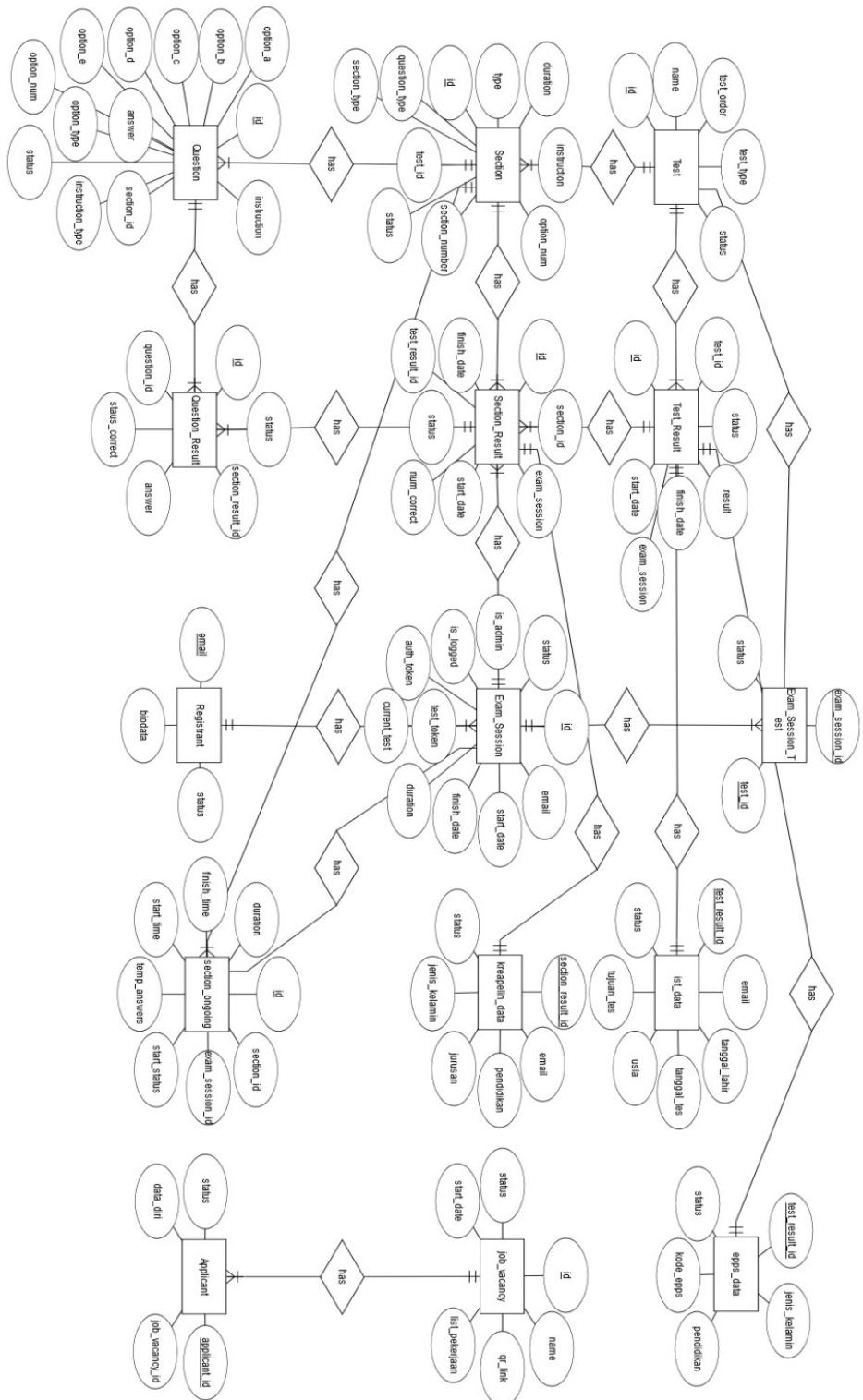
Berikut penjelasan mengenai fitur-fitur yang terdapat pada komponen ini.

1. Klik tombol silang untuk menutup komponen ini.
2. Isi nama dari lowongan yang ingin dibuat.
3. Isi nama posisi lowongan yang sedang dibuka.
4. Tekan tombol tambahkan untuk menambahkan posisi lowongan ke fitur nomor 5.
5. Tampilan daftar lowongan apa saja yang ditawarkan. Tekan tombol silang pada salah satu lowongan untuk menghilangkan lowongan dari daftar lowongan.
6. Tekan tombol buat untuk membuat lowongan pekerjaan baru. Program akan membuat lowongan pekerjaan baru serta membuat QR Code secara otomatis yang mengarahkan ke halaman website lowongan ini.

DAFTAR PUSTAKA

1. Sutrisni, Arum (2020). "Pengertian Teks Diskusi". [Online] Available at: <https://www.kompas.com/skola/read/2020/09/01/173000669/pengertian-teks-diskusi?page=all>. Diakses pada (31 Agustus 2021, 12:09).
2. Qosdil A. (2015)." Istilah ‘Frontend’ dan ‘Backend’ Membingungkan?”. [Online] Available at: <https://medium.com/@qosdil/istilah-front-end-dan-back-end-yang-membingungkan-7e091b99cfb6> Diakses pada (31 Agustus 2021, 12:34).
3. Adani, Muhammad Robith (2020). "Tahapan Pengembangan Perangkat Lunak dengan Metode Waterfall". [Online] Available at : <https://www.sekawanmedia.co.id/blog/metode-waterfall/> Diakses pada (23 Mei 2022, 23:50).
4. Nikolaieva, Aliona (2020). "8 Best Software Development Methodologies". [Online] Available at : <https://uptech.team/blog/software-development-methodologies> Diakses pada (31 Agustus 2021, 17:10).
5. Kurnia, Tri (2015). "Pengertian forum beserta contoh dan keterangannya". [Online] Available at : <http://trikurnia2.blogspot.com/2015/08/pengertian-forum-beserta-contoh-dan.html> Diakses pada (12 September 2021, 18:58).
6. wH1T3KN1gHt (2020). "Mengenal Karoseri Adiputro, Pioner Karoseri Bis Indonesia". [Online] Available at: <https://www.marjayatrans.com/mengenal-karoseri-adiputro-pioner-karoseri-bis-indonesia/> Diakses pada (21 Mei 2022, 13:00)

Lampiran



Lampiran 1
Entity Relation Diagram (CDM)



Institut Sains dan Teknologi Terpadu Surabaya

Jalan Ngagel Jaya Tengah 73 – 77, Surabaya 60284 , Indonesia
Telp. (031) 5027920 Fax. (031) 5041509

KEHADIRAN KERJA PRAKTEK (KP)

Semester ~~Gasal~~ / Genap *) Tahun Ajaran 2021 / 2022

Nama Mahasiswa : Lawrence Patrick

NRP Mahasiswa : 219116837

Alamat : Jalan Polak Wonorejo Gang Lebar No.6, Surabaya, Jawa Timur

Telp / HP : 082245264679

Periode Kerja Praktek : 14/01/2022 (dd/mm/yyyy) sampai 14/06/2022 (dd/mm/yyyy)

Data Perusahaan Tempat Kerja Praktek

Nama Perusahaan : PT Adiputro Wirasejati

Alamat Perusahaan : Jl. Raya Balearjosari no. 35, Malang, Jawa Timur

Contact Person : Lydia Mirawati

Telp / HP : 08986329121

Catatan khusus (Persyaratan tambahan dari pihak perusahaan):

*) Coret yang tidak perlu



Institut Sains dan Teknologi Terpadu Surabaya

Jalan Ngagel Jaya Tengah 73 – 77, Surabaya 60284 , Indonesia

Telp. (031) 5027920 Fax. (031) 5041509

LAMPIRAN KEHADIRAN KERJA PRAKTEK Semester Gasal / Genap *) Tahun Ajaran 2021 / 2022

No	Tanggal	Laporan kerja yang dihasilkan	Tanda tangan dan stempel petugas	Catatan petugas	
	Pukul				
1	7 Januari 2022	Diskusi awal dan wawancara program	P.T. ADIPUTRO WIRASEJATI <u>MALANG</u> Widia		
	08.00 WIB				
2	21 Januari 2022	Progres website program. Tampilan website dashboard dan tes normal	P.T. ADIPUTRO WIRASEJATI <u>MALANG</u> Widia		
	08.00 WIB				
3	4 Februari 2022	Progres tes tintum dan uji coba tintum dan EPPS. Tampilan admin, laporan tes, laporan biodata	P.T. ADIPUTRO WIRASEJATI <u>MALANG</u> Widia		
	13.00 WIB				
4	25 Februari 2022	Progres tes berhitung, analisa, penalaran, dll. Analisa norma-norma tes Uji coba tes EPPS	P.T. ADIPUTRO WIRASEJATI <u>MALANG</u> Widia		
	13.00 WIB				
5	1 Maret 2022	Uji coba tes berhitung, dll Analisa tes kereapelin	P.T. ADIPUTRO WIRASEJATI <u>MALANG</u> Widia		
	13.00 WIB	Analisa file excel input soal			
Catatan Dosen Pembimbing:				Tanda Tangan Dosen Pembimbing	
				Dr. Yosi Kristian,S.Kom.,M.Kom.	

*) Coret yang tidak perlu



Institut Sains dan Teknologi Terpadu Surabaya

Jalan Ngagel Jaya Tengah 73 – 77, Surabaya 60284 , Indonesia

Telp. (031) 5027920 Fax. (031) 5041509

LAMPIRAN KEHADIRAN KERJA PRAKTEK Semester Gasal / Genap *) Tahun Ajaran 2021 / 2022

No	Tanggal	Laporan kerja yang dihasilkan	Tanda tangan dan stempel petugas	Catatan petugas		
	Pukul					
1	15 Maret 2022	Analisa Rekrutmen (Lowongan Pekerjaan)	P.T. ADIPUTRO WIRASEJATI <u>M A D Y A N I</u> M A L A N G			
	13.00 WIB					
2	25 Maret 2022	Uji coba tes kreapelin Uji coba server (coba tes) Analisa tes MMPI, SDI	P.T. ADIPUTRO WIRASEJATI <u>M A D Y A N I</u> M A L A N G			
	13.00 WIB					
3	8 April 2022	Uji coba tes MMPI, SDI, rekrutmen. Analisa tes IST, Adikudag, Hafalan, Papi kostick, tintum anak, gambar	P.T. ADIPUTRO WIRASEJATI <u>M A D Y A N I</u> M A L A N G			
	09.30 WIB					
4	26 April 2022	Uji coba tes Adikudag, IST, tintumanak, hafalan, papi kostick	P.T. ADIPUTRO WIRASEJATI <u>M A D Y A N I</u> M A L A N G			
	13.00 WIB					
5	13 Mei 2022	Uji Coba tes gambar, Analisa server Adiputro Progres report Admin, upload kertas buram. Halaman web responsive	P.T. ADIPUTRO WIRASEJATI <u>M A D Y A N I</u> M A L A N G			
	09.00 WIB					
Catatan Dosen Pembimbing:				Tanda Tangan Dosen Pembimbing		
				Dr. Yosi Kristian,S.Kom.,M.Kom.		

*) Coret yang tidak perlu



Institut Sains dan Teknologi Terpadu Surabaya

Jalan Ngagel Jaya Tengah 73 – 77, Surabaya 60284 , Indonesia

Telp. (031) 5027920 Fax. (031) 5041509

KEHADIRAN KERJA PRAKTEK (KP)

Semester Gasal / Genap *) Tahun Ajaran 2021 / 2022

Nama Mahasiswa : Widean Nagari

NRP Mahasiswa : 219116863

Alamat : Suvadiva Z15-10, Grand Island, Pakuwon City, Surabaya, Jawa Timur

Telp / HP : 087853739169

Periode Kerja Praktek : 14/01/2022 (dd/mm/yyyy) sampai 14/06/2022 (dd/mm/yyyy)

Data Perusahaan Tempat Kerja Praktek

Nama Perusahaan : PT Adiputro Wirasejati

Alamat Perusahaan : Jl. Raya Balearjosari no. 35, Malang, Jawa Timur

Contact Person : Lydia Mirawati

Telp / HP : 08986329121

Catatan khusus (Persyaratan tambahan dari pihak perusahaan):

*) Coret yang tidak perlu



Institut Sains dan Teknologi Terpadu Surabaya

Jalan Ngagel Jaya Tengah 73 – 77, Surabaya 60284 , Indonesia

Telp. (031) 5027920 Fax. (031) 5041509

LAMPIRAN KEHADIRAN KERJA PRAKTEK Semester Gasal / Genap *) Tahun Ajaran 2021 / 2022

No	Tanggal	Laporan kerja yang dihasilkan	Tanda tangan dan stempel petugas	Catatan petugas
	Pukul			
1	7 Januari 2022	Diskusi awal dan wawancara program	P.T ADIPUTRO WIRASEJATI <u>Lydia</u> <u>MALANG</u>	
	08.00 WIB			
2	21 Januari 2022	Progres website program. Tampilan website dashboard dan tes normal	P.T ADIPUTRO WIRASEJATI <u>Lydia</u> <u>MALANG</u>	
	08.00 WIB			
3	4 Februari 2022	Progres tes tintum dan uji coba tintum dan EPPS. Tampilan admin, laporan tes, laporan biodata	P.T ADIPUTRO WIRASEJATI <u>Lydia</u> <u>MALANG</u>	
	13.00 WIB			
4	25 Februari 2022	Progres tes berhitung, analisa, penalaran, dll. Analisa norma-norma tes Uji coba tes EPPS	P.T ADIPUTRO WIRASEJATI <u>Lydia</u> <u>MALANG</u>	
	13.00 WIB			
5	1 Maret 2022	Uji coba tes berhitung,dll Analisa tes kereapelin	P.T ADIPUTRO WIRASEJATI <u>Lydia</u> <u>MALANG</u>	
	13.00 WIB	Analisa file excel input soal	Lydia	
Catatan Dosen Pembimbing:				Tanda Tangan Dosen Pembimbing
				Dr. Yosi Kristian,S.Kom.,M.Kom.

*) Coret yang tidak perlu



Institut Sains dan Teknologi Terpadu Surabaya

Jalan Ngagel Jaya Tengah 73 – 77, Surabaya 60284 , Indonesia

Telp. (031) 5027920 Fax. (031) 5041509

LAMPIRAN KEHADIRAN KERJA PRAKTEK Semester Gasal / Genap *) Tahun Ajaran 2021 / 2022

No	Tanggal	Laporan kerja yang dihasilkan	Tanda tangan dan stempel petugas	Catatan petugas
	Pukul			
1	15 Maret 2022	Analisa Rekrutmen (Lowongan Pekerjaan)	P.T. ADIPUTRO WIRASEJATI <u>MALANG</u> Lydia	
	13.00WIB			
2	25 Maret 2022	Uji coba tes Kreupelin Uji coba server (coba tes) Analisa tes MMPI, SDI	P.T. ADIPUTRO WIRASEJATI <u>MALANG</u> Lydia	
	13.00WIB			
3	8 April 2022	Uji coba tes MMPI, SDI, rekrutmen. Analisa tes IST, Adikudag, Hafalan, Papi kostick, tintum arak, gambar	P.T. ADIPUTRO WIRASEJATI <u>MALANG</u> Lydia	
	09.30WIB			
4	26 April 2022	Uji coba tes Adikudag, IST, tintumanak, hafalan, Papi kostick	P.T. ADIPUTRO WIRASEJATI Lydia	
	13.00WIB			
5	13 Mei 2022	Uji Coba tes gambar, Analisa server Adiputro Progres report Admin, upload kertas buram. Halaman web responsive	P.T. ADIPUTRO WIRASEJATI <u>MALANG</u> Lydia	Tanda Tangan Dosen Pembimbing
	09.00WIB			
Catatan Dosen Pembimbing:				Drs. Yosi Kristian,S.Kom.,M.Kom.

*) Coret yang tidak perlu