

LAMPIRAN

Lampiran 1. Surat Kesepakatan Bimbingan Skripsi

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan dibawah ini:

Pihak Pertama

Nama : Dinda Ari Theanekawati
NIM : 21090071
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Ir. Ginanjar Wiro Sasmito, M.Kom., IPM., ASEAN Eng.
Status : Dosen
NIDN : 0613028601
Jabatan Fungsional : Lektor Kepala
Pangkat/Golongan : Penata Tk. I/ III D

Pada hari ini Jum'at tanggal 07 Maret 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 8 kali kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.
Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi

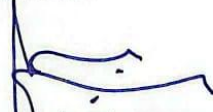
Tegal, 07 Maret 2025

Pihak Pertama



Dinda Ari Theanekawati

Pihak Kedua



Ir. Ginanjar Wiro Sasmito, M.Kom., IPM., ASEAN Eng.

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik Informatika


Dyah Aprihanti, S.T., M.Kom.
NIPY. 09.015.225

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan dibawah ini:

Pihak Pertama

Nama : Dinda Ari Theanekawati
NIM : 21090071
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Muhammad Fikri Hidayattullah, S.T., M.Kom.
Status : Dosen
NIDN : 0623108801
Jabatan Fungsional : Lektor
Pangkat/Golongan : Penata Tk. I/III-d

Pada hari ini Jum'at tanggal 07 Maret 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib menyelesaikan Skripsi maksimal pada bulan Juni 2025, jika melebihi batas waktu maka Pihak Kedua berhak untuk tidak melanjutkan proses bimbingan Skripsi. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 07 Maret 2025

Pihak Pertama



Dinda Ari Theanekawati

Pihak Kedua



Muhammad Fikri Hidayattullah, S.T., M.Kom.

Mengetahui

Ketua Program Studi Sarjana Terapan Teknik Informatika



Dyah Aprihanti, S.T., M.Kom.

NIPY-09.015.225

Lampiran 2. Surat Pernyataan Pengajuan HKI

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1. Nama : Dinda Ari Theanekawati
Kewarganegaraan : Indonesia
Alamat : Jl. Sukosrono No. 28 Slerok, Kecamatan Tegal Timur, Kota Tegal, Jawa Tengah 52125
2. Nama : Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Desa Kluwut RT.003/002, Kecamatan Bulakamba, Kabupaten Brebes, Provinsi Jawa Tengah, 52253
3. Nama : Muhammad Fikri Hidayattullah, S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jl. Glatik No. 68 Randugunting, Kota Tegal, Jawa Tengah 52131

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
Berupa : Program Komputer
Berjudul : *FFTalent: Sistem E-Recruitment dan Perangkingan Pemain Free Fire Menggunakan Metode KNN Berbasis Mobile*
 - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
 - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
 - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
 - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
 - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
 - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.
2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.
3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.

4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:
- permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
 - Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
 - Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam perkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Tegal, 26 Juni 2025



(Dinda Ari Theanekawati)

(Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom)

(Muhammad Fikri Hidayattullah, S. T., M. Kom)

* Semua pemegang hak cipta agar menandatangani di atas materai.

Lampiran 3. Surat Pengalihan HKI

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

1. Nama : Dinda Ari Theanekawati
Kewarganegaraan : Indonesia
Alamat : Jl. Sukosono No. 28 Slerok, Kecamatan Tegal Timur, Kota Tegal, Jawa Tengah 52125
2. Nama : Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Desa Kluwut RT.003/002, Kecamatan Bulakamba, Kabupaten Brebes, Provinsi Jawa Tengah, 52253
3. Nama : Muhammad Fikri Hidayattullah, S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jl. Glatik No. 68 Randugunting, Kota Tegal, Jawa Tengah 52131

Adalah **Pihak I** selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada :

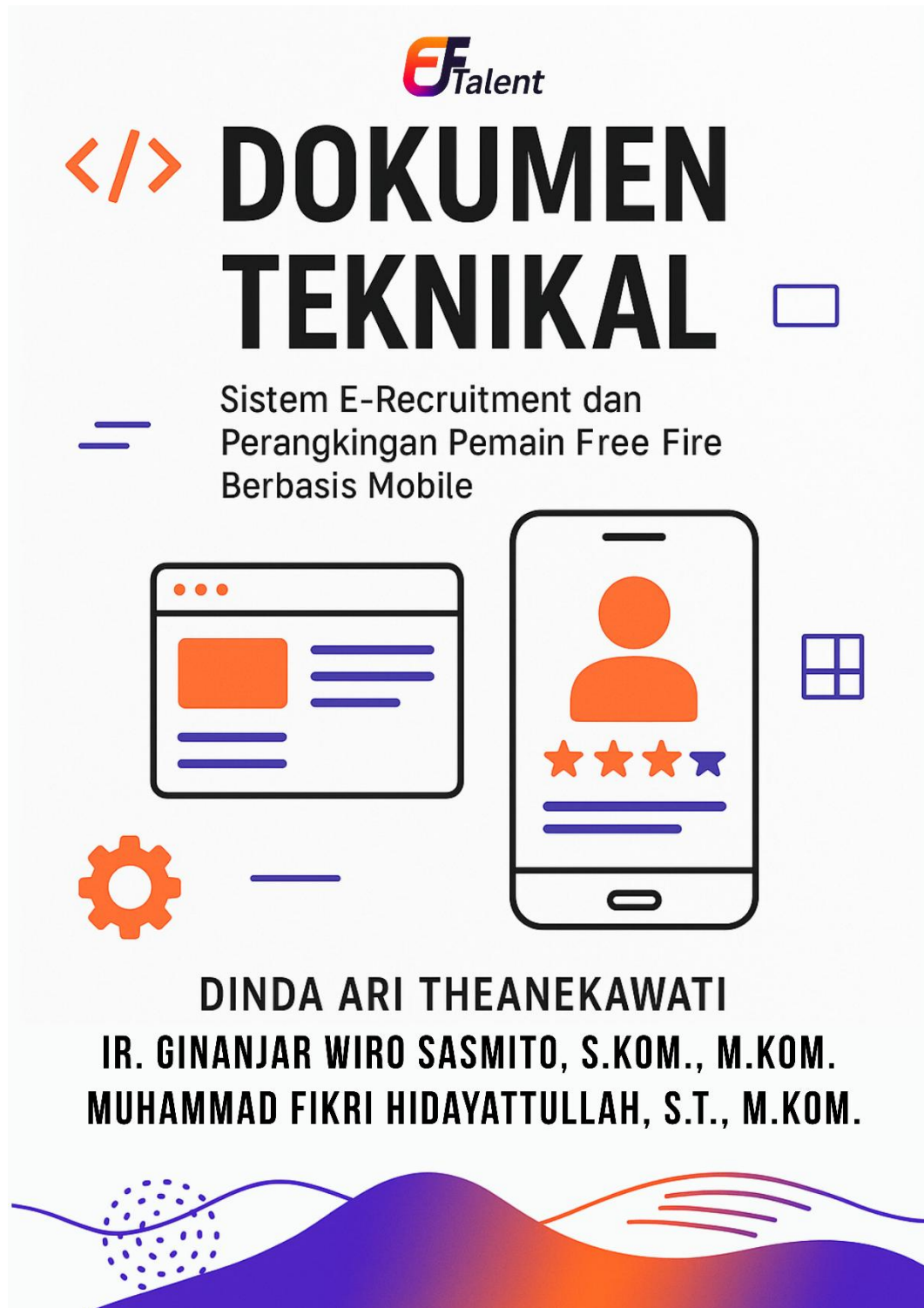
Nama : Pusat Penelitian dan Pengabdian Masyarakat (P3M)
Politeknik Harapan Bersama
Kewarganegaraan : Indonesia
Alamat : Jl. Mataram No. 9 Pesuruan Lor Kota Tegal

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "*FFTalent: Sistem E-Recruitment dan Perangkingan Pemain Free Fire Menggunakan Metode KNN Berbasis Mobile*" untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.

Tegal, 26 Juni 2025

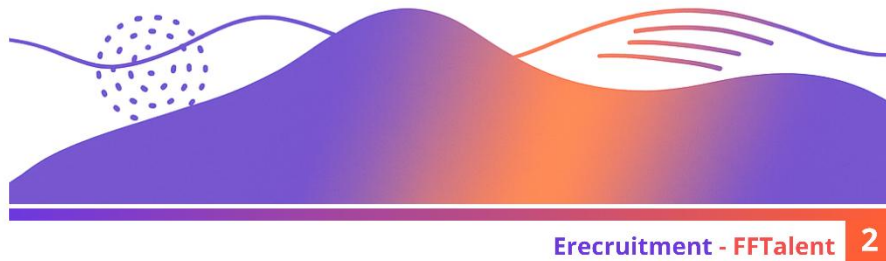
Pemegang Hak Cipta Kepala P3M  (Muhammad Fikri Hidayattullah, S.T., M. Kom)	Pencipta  (Dinda Ari Theanekawati)  (Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom)  (Muhammad Fikri Hidayattullah, S. T., M. Kom)
--	---



PENDAHULUAN

Free Fire merupakan salah satu game Battle Royale terpopuler yang telah menciptakan ekosistem e-sport aktif di Indonesia, dengan jutaan pemain dan berbagai turnamen dari tingkat lokal hingga internasional. Popularitas ini mendorong tumbuhnya kebutuhan akan pembentukan tim kompetitif yang solid. Namun, proses perekrutan pemain yang selama ini dilakukan secara manual sering kali menghadapi kendala seperti tidak efektif, memakan waktu, dan kurang objektif.

Untuk mengatasi permasalahan tersebut, dikembangkanlah aplikasi mobile Sistem E-Recruitment dan Perangkingan Pemain Free Fire. Aplikasi ini dirancang untuk memfasilitasi proses rekrutmen secara digital, transparan, dan berbasis data statistik permainan. Selain itu, sistem ini juga dilengkapi dengan fitur perangkingan otomatis menggunakan algoritma machine learning seperti K-Nearest Neighbor (K-NN), guna membantu menilai performa pemain secara lebih akurat dan objektif.



MODUL DEVELOPER

Spesifikasi teknis meliputi:

1. Modul Pengguna
2. Source Code

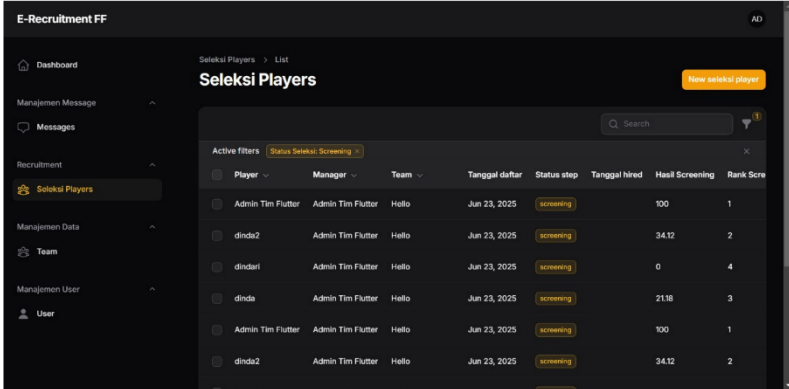
Berikut uraian spesifikasi untuk pembangunan aplikasi:

1. Windows 11 Ram 12gb
2. Visual Studio Code
3. Web Browser
4. Smartphone

Berikut uraian spesifikasi modul:

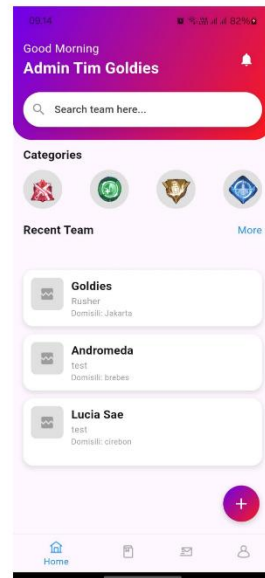
1) Modul Super Admin

Modul ini hanya dapat diakses oleh Super Admin dan berfungsi untuk mengelola serta memperbarui konten aplikasi. Fitur yang tersedia antara lain pengelolaan data tim, kategori, kriteria seleksi, serta pengawasan proses seleksi yang berjalan. Super Admin juga memiliki hak penuh dalam memantau aktivitas seluruh pengguna dan menjaga integritas sistem.



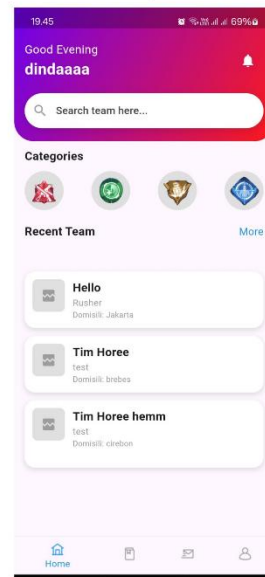
Player	Manager	Team	Tanggal daftar	Status step	Tanggal hired	Hasil Screening	Rank Score
Admin Tim Flutter	Admin Tim Flutter	Hello	Jun 23, 2025	screening		100	1
dinda2	Admin Tim Flutter	Hello	Jun 23, 2025	screening		34.12	2
dindari	Admin Tim Flutter	Hello	Jun 23, 2025	screening		0	4
dinda	Admin Tim Flutter	Hello	Jun 23, 2025	screening		21.16	3
Admin Tim Flutter	Admin Tim Flutter	Hello	Jun 23, 2025	screening		100	1
dinda2	Admin Tim Flutter	Hello	Jun 23, 2025	screening		34.12	2

2) Modul Manager



Modul ini ditujukan bagi pengguna yang bertindak sebagai manajer tim. Manager dapat membuka rekrutmen, menambahkan kriteria pencarian pemain, memantau proses seleksi, mengubah status seleksi, serta berkomunikasi langsung dengan calon pemain melalui fitur chat. Modul ini membantu manager dalam merekrut pemain yang sesuai kebutuhan tim secara efisien.

3) Modul Player



Modul ini diperuntukkan bagi pemain yang ingin mendaftar sebagai calon anggota tim. Pemain dapat membuat akun, mengisi data statistik permainan, melihat tim yang sedang membuka rekrutmen berdasarkan role yang dibutuhkan, mendaftar ke tim, memantau proses seleksi, menerima notifikasi, serta berinteraksi melalui fitur chat.

SOURCE CODE

Source code back-end dari aplikasi FFTalent menggunakan framework Laravel yang digunakan sebagai route untuk melakukan login, register, mengelola data pengguna, serta menampilkan data kriteria, tim, seleksi, dan notifikasi. Back-end ini juga mendukung proses pendaftaran tim, pengaturan tahap seleksi, serta fitur komunikasi antar pengguna. Selain itu, sistem ini menyediakan endpoint yang digunakan untuk menampilkan data pada aplikasi mobile sesuai dengan peran user, yaitu admin, manager, dan player.

1) Login

```
1 public function login(Request $request)
2 {
3     $validator = Validator::make($request->all(), [
4         'email' => 'required|email',
5         'password' => 'required'
6     ]);
7
8     if ($validator->fails()) {
9         return response()->json(['errors' => $validator->errors()], 422);
10    }
11
12    $user = User::where('email', $request->email)->first();
13
14    if (!$user || !Hash::check($request->password, $user->password)) {
15        return response()->json(['message' => 'Invalid credentials'], 401);
16    }
17
18    $token = $user->createToken('auth_token')->plainTextToken;
19
20    return response()->json([
21        'code' => 201,
22        'message' => 'Login successful.',
23        'data' => $user,
24        'access_token' => $token,
25        'token_type' => 'Bearer',
26    ], 200);
27 }
```

Fungsi login digunakan untuk menangani proses autentikasi pengguna dengan memvalidasi input email dan password yang dikirim dari form login. Jika validasi berhasil, sistem akan mencari data pengguna berdasarkan email dan memverifikasi kecocokan password menggunakan hash. Jika data valid, maka sistem akan menghasilkan token autentikasi menggunakan Laravel Sanctum dan mengembalikannya bersama data pengguna dalam response JSON. Fungsi ini memastikan hanya pengguna dengan kredensial yang benar yang dapat mengakses sistem.

2) Register

```
1 public function register(RegisterUserRequest $request)
2 {
3     $data = $request->validated();
4
5     // Buat user terlebih dahulu
6     $user = User::create([
7         'name' => $data['name'],
8         'email' => $data['email'],
9         'password' => Hash::make($data['password']),
10        'role' => $data['register_as'],
11
12        'nama_lengkap' => $data['nama_lengkap'],
13        'umur' => $data['umur'],
14        'jenis_kelamin' => $data['jenis_kelamin'],
15        'domisili' => $data['domisili'],
16        'role_player' => $data['role_player'] ?? null,
17        'nick' => $data['nick'] ?? null,
18        'free_fire_id' => $data['free_fire_id'] ?? null,
19        'rank' => $data['rank'] ?? null,
20        'rasio_kill' => $data['rasio_kill'] ?? null,
21        'rasio_headshot' => $data['rasio_headshot'] ?? null,
22        'jumlah_match' => $data['jumlah_match'] ?? null,
23        'mantan_tim_esport' => $data['mantan_tim_esport'] ?? null,
24        'skill_level' => $data['skill_level'] ?? null,
25        'tanggal_masuk' => $data['tanggal_masuk'] ?? null,
26        'tanggal_keluar' => $data['tanggal_keluar'] ?? null,
27    ]);
28
29    // Jika register_as adalah manager, buat tim
30    if ($data['register_as'] === 'manager') {
31        $teamData = [
32            'manager_id' => $user->id,
33            'nama_tim' => $data['nama_tim'] ?? 'Nama Tim Baru',
34        ];
35
36        if ($request->hasFile('logo')) {
37            $teamData['logo'] = $request->file('logo')->store('uploads', 'public');
38        }
39
40        $team = Team::create($teamData);
41
42        // update team_id user
43        $user->team_id = $team->id;
44        $user->save();
45    }
46
47    return response()->json([
48        'code' => 201,
49        'message' => 'User registered successfully.',
50        'data' => new UserResourceApi($user),
51    ], 201);
52 }
```

Fungsi register digunakan untuk menangani proses pendaftaran pengguna baru dengan terlebih dahulu memvalidasi data yang dikirim melalui RegisterUserRequest.

Setelah data valid, sistem membuat entri pengguna baru ke database dengan informasi lengkap seperti nama, email, password yang di-hash, serta detail profil lainnya. Jika pengguna mendaftar sebagai manager, maka sistem juga otomatis membuat tim baru dan mengaitkan tim tersebut ke pengguna. Setelah proses selesai, sistem akan mengembalikan respons JSON yang menyatakan bahwa pendaftaran berhasil beserta data pengguna yang telah dibuat.

3) Send Otp

```
1 public function sendOtp(Request $request)
2 {
3     $validator = Validator::make($request->all(), [
4         'email' => 'required|email|exists:users,email',
5     ]);
6
7     if ($validator->fails()) {
8         return response()->json(['errors' => $validator->errors()], 422);
9     }
10
11     $user = User::where('email', $request->email)->first();
12
13     // Generate OTP 6 digit random angka
14     $otpCode = rand(100000, 999999);
15
16     // Simpan OTP ke tabel otp_codes (pastikan sudah dibuat model dan migrasi)
17     OtpCode::updateOrCreate(
18         ['email' => $user->email],
19         [
20             'code' => $otpCode,
21             'expires_at' => Carbon::now()->addMinutes(5) // OTP valid 5 menit
22         ]
23     );
24
25     // Kirim OTP via email
26     Mail::raw("Your OTP code is: $otpCode", function ($message) use ($user) {
27         $message->to($user->email)
28             ->subject('Your Login OTP Code');
29     });
30
31     return response()->json([
32         'message' => 'OTP has been sent to your email.'
33     ], 200);
34 }
```

Fungsi sendOtp digunakan untuk mengirimkan kode OTP (One-Time Password) ke email pengguna sebagai langkah verifikasi. Fungsi ini memvalidasi apakah email yang dimasukkan sudah terdaftar di sistem. Jika valid, sistem akan menghasilkan kode OTP acak 6 digit dan menyimpannya ke dalam tabel otp_codes beserta waktu kedaluwarsa selama 5 menit. Setelah itu, kode OTP dikirim ke email pengguna menggunakan layanan email bawaan Laravel. Fungsi ini akan merespons dengan pesan bahwa OTP telah berhasil dikirim.

4) Verify Otp

```
1 public function verifyOtp(Request $request)
2 {
3     $validator = Validator::make($request->all(), [
4         'email' => 'required|email|exists:users,email',
5         'otp' => 'required|digits:6',
6     ]);
7
8     if ($validator->fails()) {
9         return response()->json(['errors' => $validator->errors()], 422);
10    }
11
12    $otpData = OtpCode::where('email', $request->email)
13        ->where('code', $request->otp)
14        ->first();
15
16    if (!$otpData) {
17        return response()->json(['message' => 'Invalid OTP.'], 401);
18    }
19
20    if (Carbon::now()->gt($otpData->expires_at)) {
21        return response()->json(['message' => 'OTP expired.'], 401);
22    }
23
24    // OTP valid, login user
25    $user = User::where('email', $request->email)->first();
26
27    // Delete OTP setelah dipakai
28    $otpData->delete();
29
30    $token = $user->createToken('auth_token')->plainTextToken;
31
32    return response()->json([
33        'message' => 'Login successful with OTP.',
34        'user' => $user,
35        'access_token' => $token,
36        'token_type' => 'Bearer',
37    ], 200);
38 }
```

Fungsi `verifyOtp` digunakan untuk memverifikasi kode OTP yang dikirim oleh pengguna melalui email. Fungsi ini memvalidasi email dan kode OTP 6 digit yang dimasukkan, lalu mencocokkannya dengan data yang tersimpan di tabel `otp_codes`. Jika data tidak cocok atau OTP telah kedaluwarsa, maka akan dikembalikan respons gagal. Jika OTP valid, pengguna akan dianggap berhasil login, token autentikasi akan dibuat, dan OTP yang sudah digunakan akan dihapus dari database.

5) Log Out

```
1 public function logout(Request $request)
2 {
3     $request->user()->currentAccessToken()->delete();
4
5     return response()->json(['message' => 'Logged out successfully']);
6 }
```

Fungsi logout digunakan untuk menghapus token autentikasi pengguna yang sedang aktif, sehingga pengguna dianggap keluar dari sesi login. Dengan menghapus token tersebut, akses ke endpoint yang memerlukan autentikasi akan diblokir hingga pengguna login kembali. Setelah proses logout berhasil, sistem akan mengembalikan respons JSON dengan pesan bahwa logout berhasil dilakukan.

6) Route Autentifikasi

```
1 Route::post('/register', [AuthController::class, 'register']);
2 Route::post('/login', [AuthController::class, 'login']);
3 Route::post('/send-otp', [AuthController::class, 'sendOtp']);
4 Route::post('/verify-otp', [AuthController::class, 'verifyOtp']);
5
6 Route::middleware('auth:sanctum')->group(function () {
7     Route::post('/logout', [AuthController::class, 'logout']);
8 });
```

Kode ini berisi route untuk proses autentikasi pada aplikasi. '/register' menangani pendaftaran pengguna baru, '/login' untuk login menggunakan email dan password, '/send-otp' untuk mengirimkan kode OTP ke email, dan '/verify-otp' untuk memverifikasi OTP tersebut. Sementara itu, /logout dibungkus dalam middleware auth:sanctum, artinya hanya bisa diakses oleh pengguna yang sudah login dan memiliki token autentikasi aktif.

7) Seleksi

```
1 public function screening()
2 {
3     $manager = Auth::user();
4
5     if (!$manager) {
6         return response()->json(['message' => 'User belum login'], 401);
7     }
8
9     // Ambil semua user dengan role player
10    $players = User::where('role', 'player')->get();
11
12    if ($players->isEmpty()) {
13        return response()->json(['message' => 'Tidak ada player yang ditemukan'], 404);
14    }
15
16    // Siapkan payload untuk dikirim ke sistem screening
17    $payload = $players->map(function ($player) {
18        return [
19            'nama_lengkap' => $player->nama_lengkap,
20            'umur' => $player->umur,
21            'jenis_kelamin' => $player->jenis_kelamin,
22            'domisili' => $player->domisili,
23            'role' => $player->role,
24            'nick' => $player->nick,
25            'free_fire_id' => $player->free_fire_id ?? 1,
26            'rank' => $player->rank ?? 1,
27            'ratio_kill' => $player->ratio_kill ?? 1,
28            'ratio_headshot' => $player->ratio_headshot ?? 1,
29            'jumlah_match' => $player->jumlah_match ?? 1,
30            'mantan_tin_export' => $player->mantan_tin_export ?? 1,
31            'skill_level' => $player->skill_level ?? 1,
32        ];
33    })->toArray();
34
35    try {
36        $client = new GuzzleHttp\Client();
37        $response = $client->post('http://localhost:5000/predict', [
38            'json' => $payload,
39            'headers' => ['Accept' => 'application/json'],
40        ]);
41
42        $result = json_decode($response->getBody(), true);
43
44        foreach ($result as $res) {
45            $player = User::where('nama_lengkap', $res['nama_lengkap'])->first();
46            if ($player) {
47                $seleksiPlayer::create([
48                    'player_id' => $player->id,
49                    'status_step' => 'screening',
50                    'tanggal_daftar' => now(),
51                    'notes' => 'sudah screening by system',
52                    'accept_screening' => 'waiting',
53                    'tanggal_hired' => null,
54                    'manager_id' => $manager->id,
55                    'team_id' => $manager->team_id,
56                    'nilai' => $res['nilai'] ?? null,
57                    'peringkat' => $res['peringkat'] ?? null,
58                ]);
59            }
60        }
61
62        return response()->json(['message' => 'Hasil screening berhasil dikirim', 'results' => $result], 201);
63    } catch (Exception $e) {
64        return response()->json(['error' => 'Gagal memproses prediksi', 'details' => $e->getMessage()], 500);
65    }
66 }
```

Fungsi screening digunakan oleh manajer untuk melakukan proses seleksi awal terhadap semua pemain yang terdaftar dengan role player. Pertama, fungsi memverifikasi apakah pengguna (manajer) sudah login. Jika tidak, akan mengembalikan pesan error. Selanjutnya, sistem mengambil seluruh data pemain dan membentuk payload berisi atribut penting seperti nama, umur, rank, rasio kill, dan lainnya. Data ini kemudian dikirim ke endpoint Flask (<http://localhost:5000/predict>) untuk dianalisis oleh model machine learning. Hasil dari analisis tersebut—berupa nilai dan peringkat—akan disimpan ke dalam tabel seleksi_players, mencatat bahwa pemain telah melalui tahap screening secara otomatis oleh sistem. Jika terjadi error saat komunikasi dengan server Flask, maka sistem akan menangkap dan mengembalikan pesan kesalahan.

5) Api Service Flutter

```
1 class AppServices {
2   static get role => null;
3
4   static String getBaseUrl() {
5     return 'http://127.0.0.1:8001/api';
6   }
7
8 }
```

Kelas AppServices ini berfungsi sebagai utilitas untuk mendapatkan berbagai endpoint API dalam aplikasi. Berikut adalah penjelasan singkat tentang setiap metode dalam kelas ini:

- `getBaseUrl()`: Metode ini mengembalikan URL dasar dari aplikasi, dalam hal ini 'http://127.0.0.1:8001/api'.

```
1 // ===== AUTH =====
2 static String getLoginEndpoint() {
3   return '${getBaseUrl()}/login';
4 }
5
6 static String getLogoutEndpoint() {
7   return '${getBaseUrl()}/logout';
8 }
9
10 static String getRegistEndpoint() {
11   return '${getBaseUrl()}/register';
12 }
13
14 static String getVerifotpEndpoint() {
15   return '${getBaseUrl()}/verify-otp';
16 }
17
18 static String getResendotpEndpoint() {
19   return '${getBaseUrl()}/send-otp';
20 }
```

- `getLoginEndpoint()`: Metode ini mengembalikan URL endpoint untuk login, yaitu gabungan dari URL dasar dan `/login`.
- `getLogoutEndpoint()`: Metode ini mengembalikan URL endpoint untuk logout, yaitu gabungan dari URL dasar dan `/logout`.
- `getRegistEndpoint()`: Metode ini mengembalikan URL endpoint untuk registrasi pengguna baru, yaitu gabungan dari URL dasar dan `/register`.
- `getVerifotpEndpoint()`: Metode ini mengembalikan URL endpoint untuk verifikasi kode OTP, yaitu gabungan dari URL dasar dan `/verify-otp`.
- `getResendotpEndpoint()`: Metode ini mengembalikan URL endpoint untuk mengirim ulang kode OTP ke email pengguna, yaitu gabungan dari URL dasar dan `/send-otp`.

```

1 // ===== PROFILE =====
2 static String getProfileEndpoint() {
3     return '${getBaseUrl()}/my/profile';
4 }
5
6 static String getUpdateProfileEndpoint() {
7     return '${getBaseUrl()}/my/profile/update';
8 }

```

- `getProfileEndpoint()`: Metode ini mengembalikan URL endpoint untuk menampilkan profil pengguna, yaitu gabungan dari URL dasar dan `/my/profile`.
- `getUpdateProfileEndpoint()`: Metode ini mengembalikan URL endpoint untuk memperbarui data profil pengguna, yaitu gabungan dari URL dasar dan `/my/profile/update`.

```

1 // ===== PLAYER - OPREC TEAM =====
2 static String getTeamEndpoint() {
3     return '${getBaseUrl()}/player/oprec-team';
4 }
5
6 static String getRoleEndpoint(String role) {
7     return '${getBaseUrl()}/player/oprec-team/filter-by-role?kriteria_role=$role';
8 }
9
10 static String getTeamDetailEndpoint(int id) {
11     return '${getBaseUrl()}/player/oprec-team/${id}';
12 }
13
14 static String getApplyEndpoint(int id) {
15     return '${getBaseUrl()}/player/oprec-team/apply/${id}';
16 }

```

- `getTeamEndpoint()`: Metode ini mengembalikan URL endpoint untuk mendapatkan daftar tim yang membuka rekrutmen, yaitu gabungan dari URL dasar dan `/player/oprec-team`.
- `getRoleEndpoint(String role)`: Metode ini mengembalikan URL endpoint untuk menyaring tim berdasarkan role, yaitu gabungan dari URL dasar dan `/player/oprec-team/filter-by-role` dengan parameter `kriteria_role`.
- `getTeamDetailEndpoint(int id)`: Metode ini mengembalikan URL endpoint untuk melihat detail dari tim berdasarkan ID, yaitu gabungan dari URL dasar dan `/player/oprec-team/{id}`.
- `getApplyEndpoint(int id)`: Metode ini mengembalikan URL endpoint untuk mendaftar ke dalam tim berdasarkan ID tim, yaitu gabungan dari URL dasar dan `/player/oprec-team/apply/{id}`.

```

1 // ===== PLAYER - MY SELECTION =====
2 static String getMySelectionEndpoint() {
3     return '${getBaseUrl()}/player/my-selection';
4 }
5
6 static String getMySelectionDetailEndpoint(int id) {
7     return '${getBaseUrl()}/player/my-selection/${id}';
8 }

```

- `getMySelectionEndpoint()`: Metode ini mengembalikan URL endpoint untuk melihat seluruh seleksi yang diikuti oleh pemain, yaitu gabungan dari URL dasar dan `/player/my-selection`.
- `getMySelectionDetailEndpoint(int id)`: Metode ini mengembalikan URL endpoint untuk melihat detail dari seleksi yang diikuti oleh pemain, berdasarkan ID, yaitu gabungan dari URL dasar dan `/player/my-selection/{id}`.

```

1 // ===== MANAGER - SELECTION =====
2
3 static String getScreeningEndpoint() {
4     return '${getBaseUrl()}/manager/selection/screening';
5 }
6
7 static String getManagerSelectionEndpoint() {
8     return '${getBaseUrl()}/manager/selection';
9 }
10
11 static String getManagerSelectionDetailEndpoint(String playerId) {
12     return '${getBaseUrl()}/manager/selection/${playerId}';
13 }
14
15 static String getManagerSelectionDetailUpdateEndpoint(String selectionId) {
16     return '${getBaseUrl()}/manager/selection/update/${selectionId}';
17 }

```

- `getScreeningEndpoint()`: Metode ini mengembalikan URL endpoint untuk melakukan proses screening seleksi oleh manajer, yaitu gabungan dari URL dasar dan `/manager/selection/screening`.
- `getManagerSelectionEndpoint()`: Metode ini mengembalikan URL endpoint untuk melihat daftar seleksi yang dilakukan oleh manajer, yaitu gabungan dari URL dasar dan `/manager/selection`.
- `getManagerSelectionDetailEndpoint(String playerId)`: Metode ini mengembalikan URL endpoint untuk melihat detail seleksi pemain berdasarkan ID pemain, yaitu gabungan dari URL dasar dan `/manager/selection/{playerId}`.

-
- `getManagerSelectionDetailUpdateEndpoint(String selectionId)`: Metode ini mengembalikan URL endpoint untuk memperbarui data hasil seleksi berdasarkan ID seleksi, yaitu gabungan dari URL dasar dan `/manager/selection/update/{selectionId}`.

```
1 // ===== MANAGER - OPREC TEAM =====
2 static String getManagerOprecListEndpoint() {
3     return "${getBaseUrl()}/manager/oprec-team";
4 }
5
6 static String getManagerOprecDetailEndpoint(int id) {
7     return "${getBaseUrl()}/manager/oprec-team/${id}";
8 }
9
10 static String getManagerOprecStoreEndpoint() {
11     return "${getBaseUrl()}/manager/oprec-team/store";
12 }
13
14 static String getManagerOprecUpdateEndpoint(int id) {
15     return "${getBaseUrl()}/manager/oprec-team/update/${id}";
16 }
17
18 static String getManagerOprecDeleteEndpoint(int id) {
19     return "${getBaseUrl()}/manager/oprec-team/delete/${id}";
20 }
```

- `getManagerOprecListEndpoint()`: Metode ini mengembalikan URL endpoint untuk mendapatkan daftar open recruitment yang dibuat oleh manajer, yaitu gabungan dari URL dasar dan `/manager/oprec-team`.
- `getManagerOprecDetailEndpoint(int id)`: Metode ini mengembalikan URL endpoint untuk melihat detail dari open recruitment tim berdasarkan ID, yaitu gabungan dari URL dasar dan `/manager/oprec-team/{id}`.
- `getManagerOprecStoreEndpoint()`: Metode ini mengembalikan URL endpoint untuk menyimpan data open recruitment baru oleh manajer, yaitu gabungan dari URL dasar dan `/manager/oprec-team/store`.
- `getManagerOprecUpdateEndpoint(int id)`: Metode ini mengembalikan URL endpoint untuk memperbarui data open recruitment berdasarkan ID, yaitu gabungan dari URL dasar dan `/manager/oprec-team/update/{id}`.
- `getManagerOprecDeleteEndpoint(int id)`: Metode ini mengembalikan URL endpoint untuk menghapus data open recruitment berdasarkan ID, yaitu gabungan dari URL dasar dan `/manager/oprec-team/delete/{id}`.

```

1 // ===== NOTIFICATION =====
2 static String getNotificationEndpoint() {
3     return '${getBaseUrl()}/notification';
4 }

```

- `getNotificationEndpoint()`: Metode ini mengembalikan URL endpoint untuk mendapatkan notifikasi sistem, yaitu gabungan dari URL dasar dan `/notification`.

```

1 // ===== MESSAGE =====
2 static String getMessageEndpoint() {
3     return '${getBaseUrl()}/messages';
4 }
5
6 static String getMessageRoomEndpoint(int id) {
7     return '${getBaseUrl()}/messages/room-message/${id}';
8 }
9
10 static String getSendMessageEndpoint() {
11     return '${getBaseUrl()}/messages/send';
12 }
13
14 static String deleteMessageEndpoint(int id) {
15     return '${getBaseUrl()}/messages/delete/${id}';
16 }
17 }

```

- `getMessageEndpoint()`: Metode ini mengembalikan URL endpoint untuk mengakses data pesan (chat), yaitu gabungan dari URL dasar dan `/messages`.
- `getMessageRoomEndpoint(int id)`: Metode ini mengembalikan URL endpoint untuk melihat isi pesan dari sebuah ruang pesan berdasarkan ID, yaitu gabungan dari URL dasar dan `/messages/room-message/{id}`.
- `getSendMessageEndpoint()`: Metode ini mengembalikan URL endpoint untuk mengirimkan pesan baru, yaitu gabungan dari URL dasar dan `/messages/send`.
- `deleteMessageEndpoint(int id)`: Metode ini mengembalikan URL endpoint untuk menghapus pesan berdasarkan ID, yaitu gabungan dari URL dasar dan `/messages/delete/{id}`.



MANUAL BOOK

SISTEM E-RECRUITMENT
DAN PERANGKINGAN PEMAIN
FREE FIRE BERBASIS MOBILE



DINDA ARI THEANEKAWATI
IR. GINANJAR WIRO SASMITO, S.KOM., M.KOM.
MUHAMMAD FIKRI HIDAYATTULLAH, S.T., M.KOM.



DAFTAR ISI

Daftar Isi.....	2
Daftar Gambar.....	3
Pendahuluan.....	4
Tampilan Awal.....	5
Penggunaan Aplikasi.....	6
1. Registrasi.....	6
2. Verifikasi.....	7
3. Login.....	8
4. Dashboard.....	9
5. Pendaftaran Tim.....	10
6. Seleksi Pemain.....	11
7. Chat.....	12
8. Notifikasi.....	13
9. Profil.....	14
10. Logout.....	15
Pemecahan Masalah.....	16

DAFTAR GAMBAR

Gambar 1. Logo FFTalent	4
Gambar 2. Tampilan Awal.....	5
Gambar 3. Tampilan Halaman Registrasi.....	6
Gambar 4. Tampilan Halaman Verifikasi.....	7
Gambar 5. Tampilan Halaman Login.....	8
Gambar 6. Halaman Dashboard Pengguna	9
Gambar 7. Form Pendaftaran Tim	10
Gambar 8. Hasil Pencarian Tim dan Filter Role	11
Gambar 9. Detail Tim.....	12
Gambar 10. Tampilan Fitur Seleksi	13
Gambar 11. Tampilan Halaman Chat	14
Gambar 12. Tampilan Halaman Notifikasi	15
Gambar 13. Tampilan Halaman Profil dan Riwayat Tim	16
Gambar 14. Tampilan Fitur Logout	17

PENDAHULUAN

Aplikasi Sistem E-Recruitment dan Perangkingan Pemain Kompetitif Free Fire Berbasis Mobile dirancang untuk mempermudah proses perekrutan pemain e-sport secara digital dan efisien. Dengan semakin berkembangnya dunia e-sport, khususnya pada game Free Fire, kebutuhan akan sistem seleksi pemain yang objektif dan transparan menjadi sangat penting.

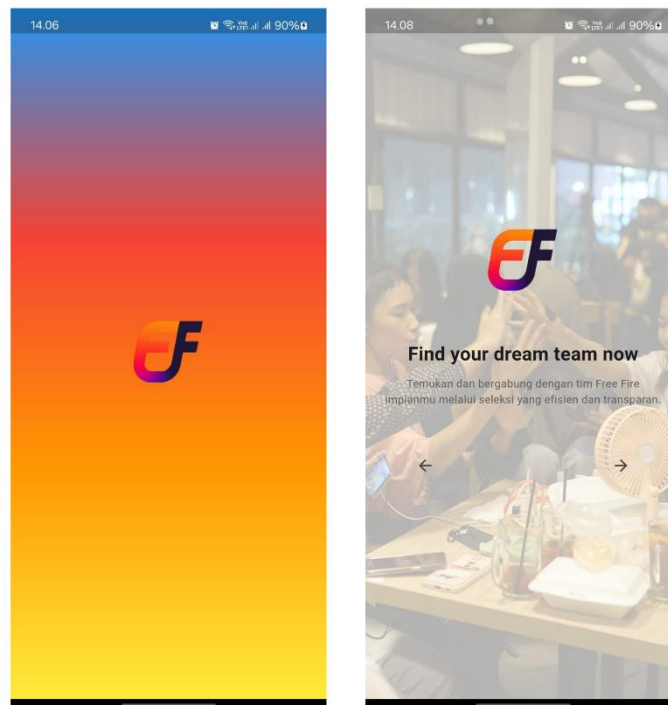
Manual book ini disusun sebagai panduan bagi pengguna aplikasi, baik itu user (pemain), admin tim (manajer tim), maupun admin sistem, untuk memahami cara penggunaan fitur-fitur yang tersedia. Di dalamnya mencakup informasi langkah-langkah registrasi, navigasi aplikasi, proses seleksi, komunikasi antar pengguna, serta penggunaan teknologi machine learning dalam perangkingan pemain.



Gambar 1. Logo FFTalent

TAMPILAN AWAL

Saat Anda pertama kali membuka aplikasi FFTalent, Anda akan melihat Splash Screen, yaitu tampilan awal berupa logo yang ditampilkan selama beberapa detik. Tampilan ini berfungsi sebagai pengenalan awal sekaligus memberikan waktu bagi sistem untuk melakukan proses inisialisasi. Setelah Splash Screen, Anda akan diarahkan ke halaman Login ketika sudah mempunyai akun.

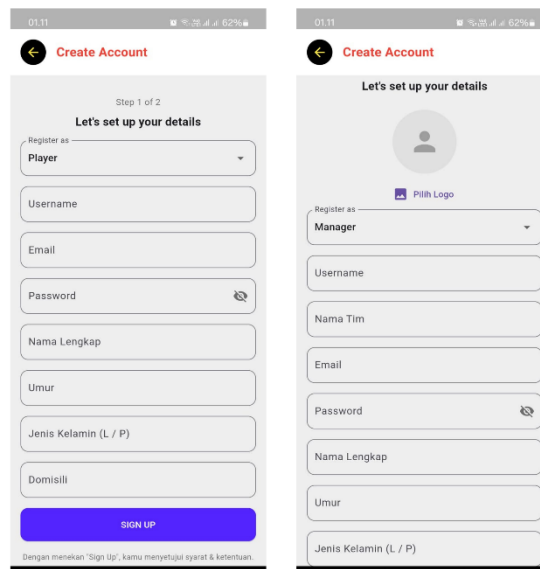


Gambar 3. Tampilan Awal

PENGUNAAN APLIKASI

Registrasi

- 1) Buka Aplikasi "E-recruitment FF Talent".
Pastikan Anda sudah menginstal aplikasi di perangkat mobile Anda.
- 2) Pilih "SignUp" untuk membuat akun baru.
Anda akan diarahkan ke halaman pendaftaran
- 3) Pilih peran Anda: "Player" atau "Manager".
Jika Anda ingin mendaftar sebagai Pemain, Pilih "Player".
Jika Anda ingin mendaftar sebagai Admin Tim, Pilih "Manager".
- 4) Isi formulir pendaftaran dengan informasi yang diminta seperti nama, email password.
- 5) Klik "SignUp" untuk menyelesaikan proses pendaftaran.
- 6) Verifikasi Email.
Sistem akan mengirimkan kode Verifikasi ke Email Anda untuk mengaktifkan akun.
- 7) Setelah berhasil mendaftar, Anda akan diarahkan ke halaman Login / SignIn untuk masuk ke aplikasi.

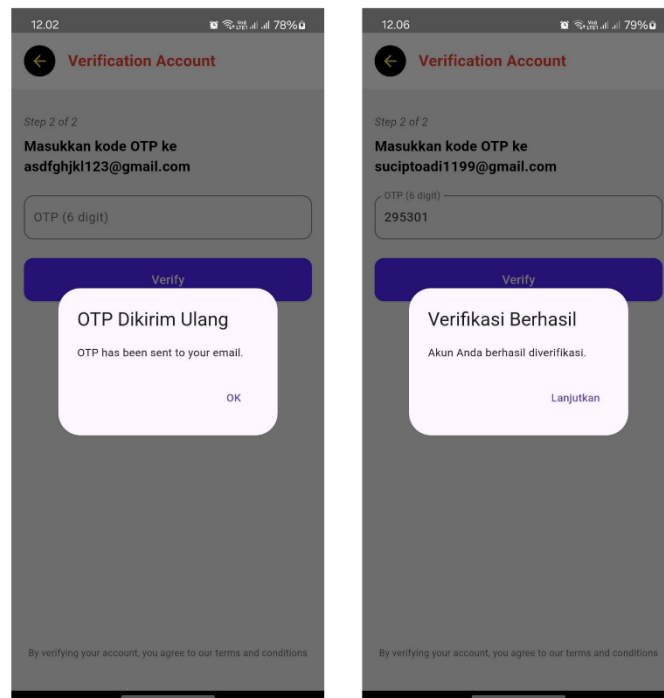


Gambar 4. Register

Verifikasi

Halaman Verifikasi digunakan untuk memastikan bahwa email yang digunakan saat registrasi benar dan aktif. Setelah berhasil mendaftar, sistem akan mengirimkan kode OTP (One-Time Password) ke alamat email yang didaftarkan.

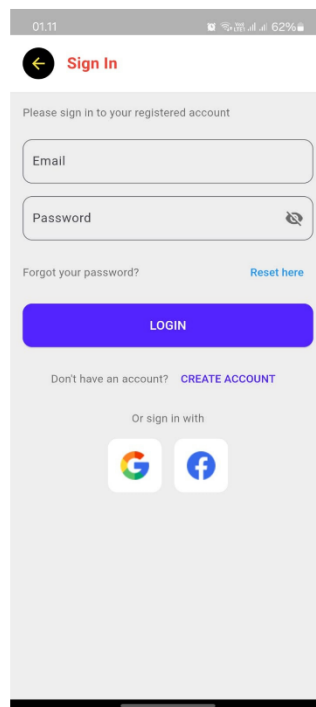
Pengguna diminta untuk memasukkan kode OTP tersebut pada kolom yang tersedia. Jika belum menerima kode, pengguna dapat menekan tombol "Resend OTP" untuk mengirim ulang kode verifikasi ke email. Proses ini bertujuan untuk menjaga keamanan dan validitas akun sebelum dapat mengakses aplikasi secara penuh.



Gambar 4. Verifikasi

Login

- 1) Buka Aplikasi "E-recruitment FF Talent".
Pastikan aplikasi sudah terpasang di perangkat Anda.
- 2) Pilih "SignIn".
- 3) Masukkan Email dan Password yang sudah di Daftar.
- 4) Klik "SignIn" untuk masuk ke akun anda.
Otomatis akan diarahkan ke halaman Dashboard sesuai role (Player atau Manager).



Gambar 5. Login

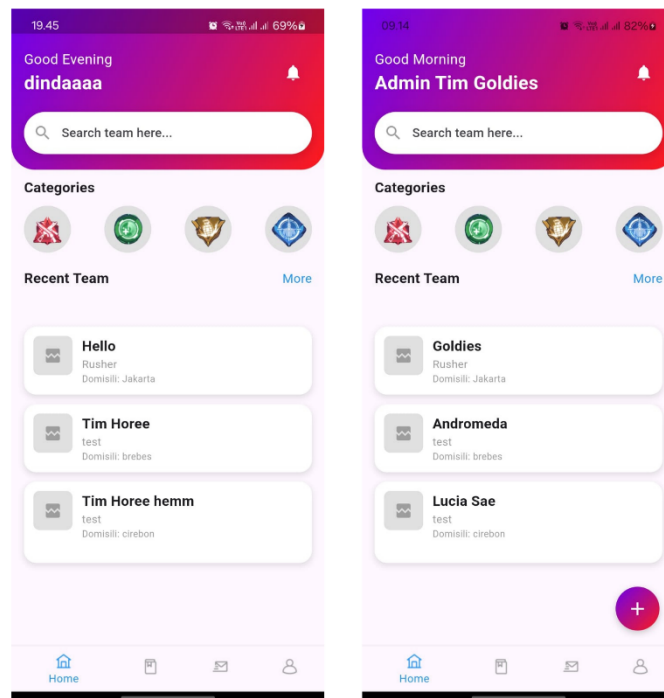
Dashboard

Setelah berhasil melakukan Sign In, pengguna akan diarahkan ke halaman Dashboard sebagai beranda utama aplikasi. Di halaman ini, pengguna dapat melihat daftar tim yang sedang membuka rekrutmen dan melakukan pendaftaran sesuai minat dan kemampuan. Masing-masing tim menampilkan detail kriteria yang dibutuhkan, sehingga pengguna dapat memahami persyaratan sebelum bergabung.

Tersedia Navbar di bagian bawah aplikasi untuk mengakses fitur utama seperti Seleksi, Chat, Profil. Pengguna juga dapat memanfaatkan fitur filter berdasarkan role pemain (seperti Rusher, Support, Bomber, atau Sniper) untuk menyesuaikan pencarian tim dengan posisi yang diinginkan.

Pada Manager, dashboard menyediakan fitur tambahan seperti:

- Menambahkan kriteria pencarian pemain saat membuka rekrutmen.
- Melihat pendaftar yang masuk, serta mengedit atau menghapus kriteria tim sesuai kebutuhan.

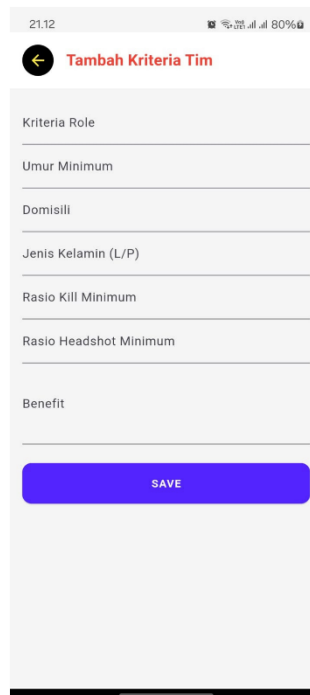


Gambar 4. Dashboard

Tambah Kriteria Rekrutmen

Manager dapat menambahkan kriteria spesifik saat membuka rekrutmen untuk menentukan tipe pemain yang dibutuhkan oleh tim. Kriteria yang dapat ditentukan meliputi role pemain (seperti Rusher, Support, Bomber, atau Sniper), Jenis Kelamin, Minimum Umur, dll.

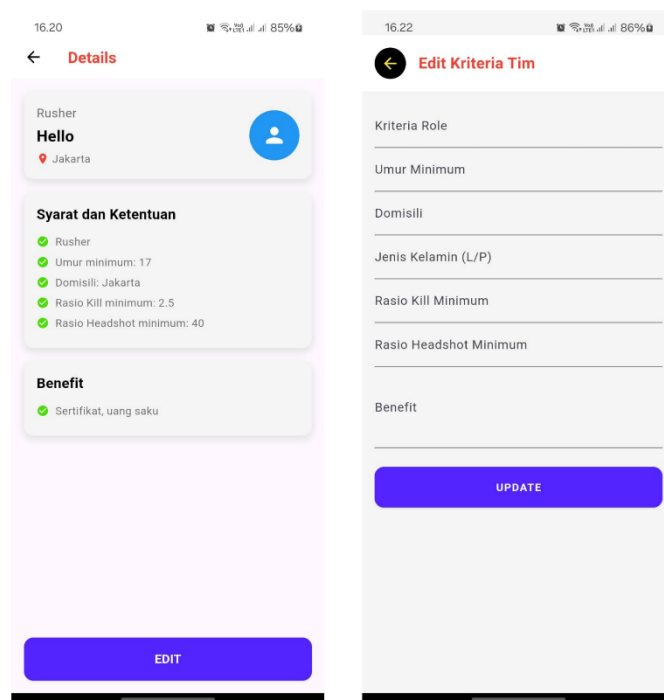
Fitur ini membantu proses seleksi menjadi lebih terarah, objektif, dan efisien, sehingga hanya pemain yang memenuhi kriteria yang akan mendaftar ke tim. Kriteria yang telah ditambahkan akan ditampilkan pada halaman detail tim dan bisa disesuaikan kembali sewaktu-waktu.



Gambar 5. Tambah Kriteria Rekrutmen

Detail Tim Manager

Halaman ini menampilkan informasi lengkap mengenai tim yang dikelola oleh manager, termasuk daftar kriteria pemain yang sedang dibutuhkan. Melalui halaman ini, manager dapat melihat, mengedit, maupun menghapus kriteria rekrutmen yang sudah ditambahkan sebelumnya.

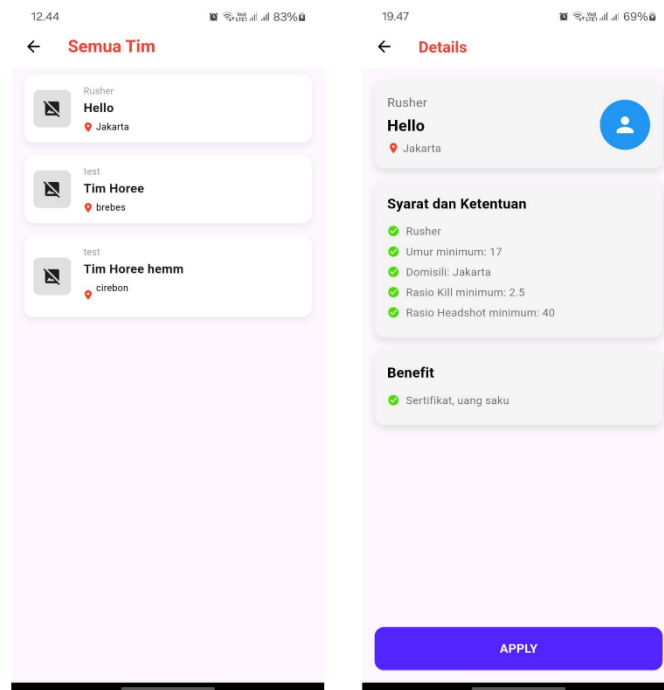


Gambar 6. Detail Tim Manager

Detail Tim Player

Pemain dapat mendaftar ke tim-tim yang sedang membuka rekrutmen melalui halaman detail tim. Berikut langkah-langkah untuk mengakses dan melakukan pendaftaran:

- 1) Lihat daftar tim yang sedang membuka rekrutmen pada Dashboard.
- 2) Pilih salah satu tim yang sesuai dengan minat Anda, lalu klik untuk melihat detail tim.
- 3) Jika sesuai, klik tombol “Apply” untuk mendaftar sebagai peserta seleksi tim tersebut.
- 4) Tunggu proses seleksi oleh tim.

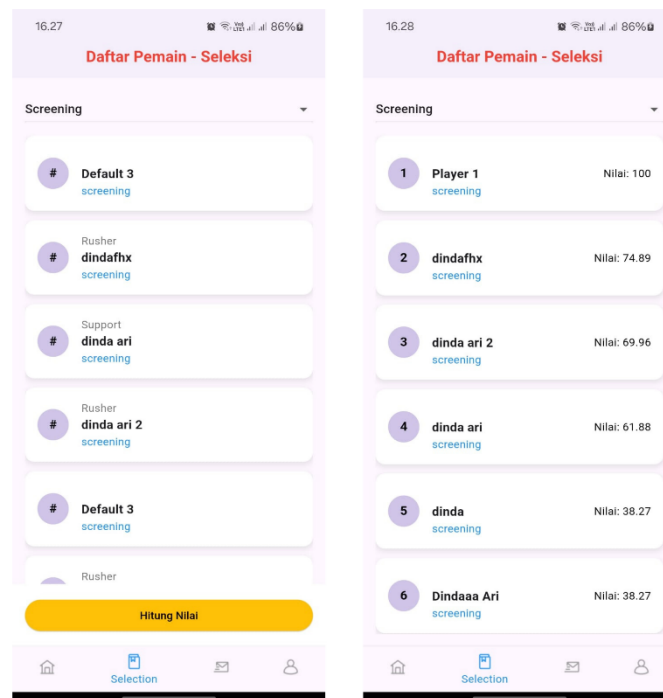


Gambar 7. Detail Tim Player

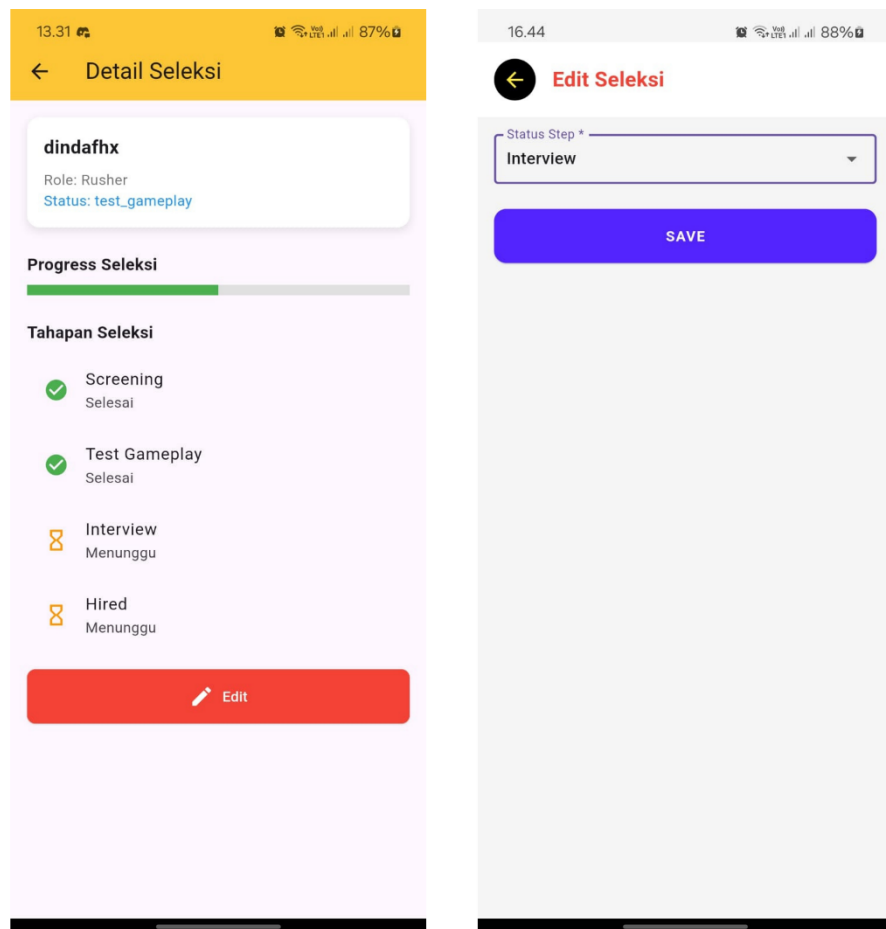
Seleksi Pemain Manager

Manager dapat mengelola proses seleksi pemain secara menyeluruh melalui menu Seleksi, termasuk melihat daftar pendaftar, memfilter berdasarkan tahap seleksi, serta mengedit status masing-masing pemain.

- 1) Buka menu Seleksi pada navbar.
 - Dapat memfilter pemain berdasarkan status tahap seleksi, seperti: Screening, Test Gameplay, Interview, atau Hired.
 - Pada tahap Screening, tersedia tombol Perangkingan untuk melihat urutan nilai pemain yang telah diakumulasi menggunakan model KNN.
- 2) Klik salah satu tim untuk melihat detail daftar pemain yang mendaftar ke tim.
 - Terdapat tombol Edit untuk mengubah status tahap seleksi tiap pemain sesuai progres, seperti dari screening ke test gameplay atau interview ke hired dll.



Gambar 8. Seleksi Pemain Manager

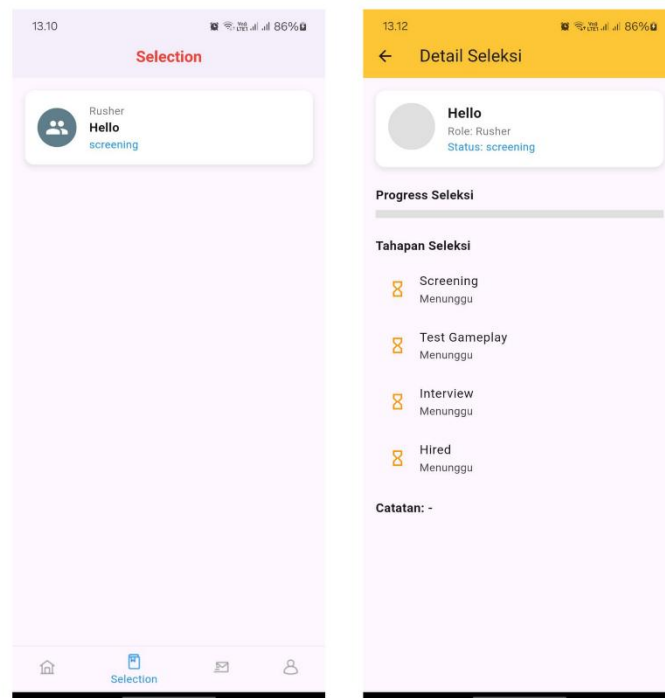


Gambar 9. Edit Seleksi Pemain Manager

Seleksi Pemain Player

Proses penyaringan pemain yang dilakukan oleh sistem dan/atau manajer tim menggunakan data statistik, tes, dan interview.

- 1) Buka menu Seleksi pada navbar.
- 2) Pilih salah satu tim yang telah Anda daftarkan untuk melihat detail seleksinya.
- 3) Anda dapat melihat status dan progres tahap seleksi.
- 4) Tunggu hingga tim memberikan hasil akhir.

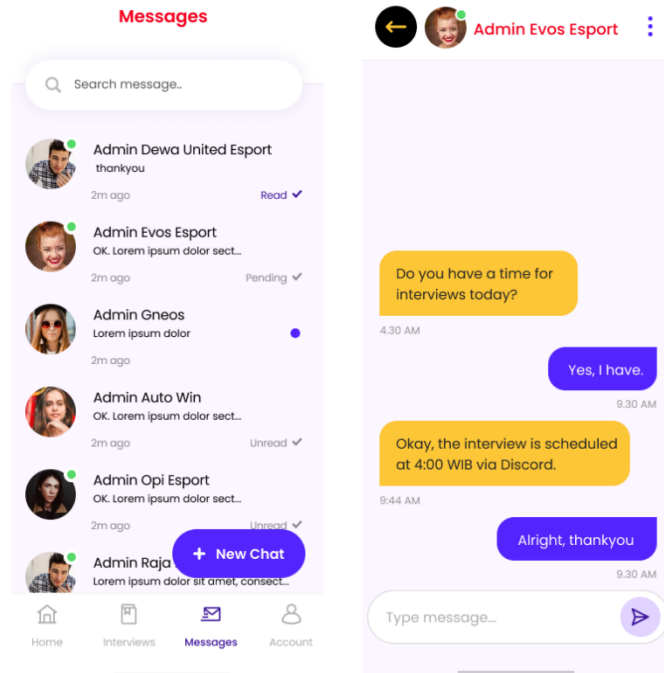


Gambar 10. Seleksi Pemain Player

Chat

Fitur Chat digunakan sebagai sarana komunikasi antara pemain dan manajer tim atau admin selama proses seleksi berlangsung. Chat dapat digunakan untuk konsultasi seleksi, jadwal, dan informasi tim.

- 1) Masuk ke menu Chat dari Navbar Dashboard.
- 2) Pilih kontak (misalnya, manajer tim).
- 3) Tulis pesan dan tekan Kirim.
- 4) Tunggu proses seleksi oleh tim.

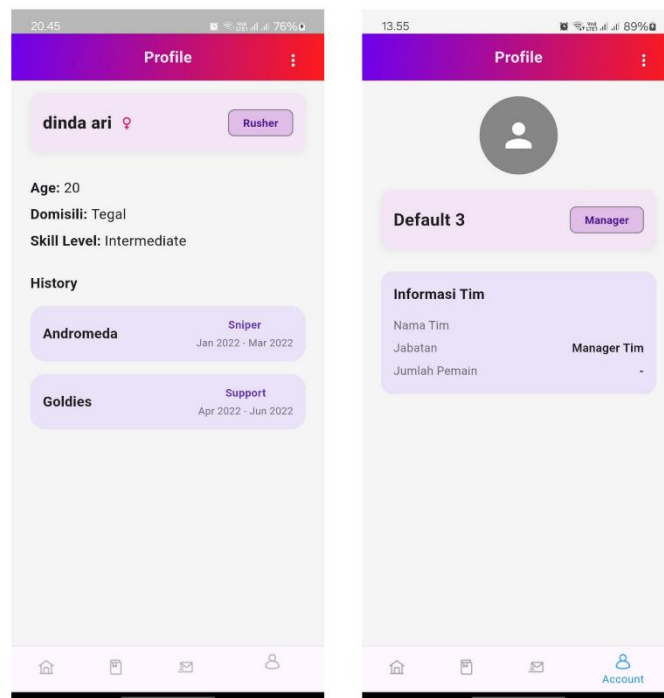


Gambar 11. Chat

Profil

Proses penyaringan pemain yang dilakukan oleh sistem dan/atau manajer tim menggunakan data statistik, tes, dan interview.

- 1) Buka menu Profil pada navbar di Beranda.
- 2) Lihat informasi pribadi seperti nama, umur, dan role player.
- 3) Player dapat melihat Riwayat Tim yang pernah diikuti.
- 4) Terdapat PopUp berisi Edit Profil, LogOut.

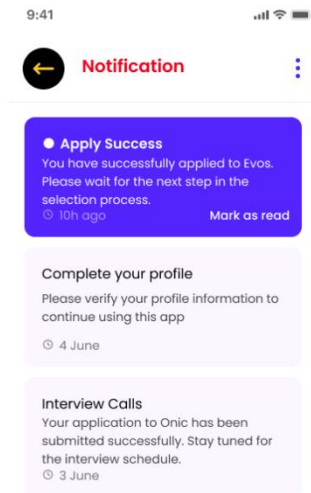


Gambar 12. Profil

Notifikasi

Fitur Notifikasi digunakan untuk menyampaikan informasi penting kepada pengguna seperti berhasil mendaftar, pengumuman hasil seleksi, atau pesan sistem lainnya.

- 1) Masuk ke menu Notifikasi yang ada di pojok kanan atas Dashboard.
- 2) Lihat daftar notifikasi yang masuk.
- 3) Notifikasi akan otomatis ditandai sebagai telah dibaca.



Gambar 13. Notifikasi

PEMECAHAN MASALAH

1) Tidak Bisa Login

- a. Periksa Koneksi Internet: Pastikan perangkat Anda terhubung ke internet.
- b. Cek Email dan Kata Sandi: Pastikan Anda memasukkan email dan kata sandi yang benar.

2) Aplikasi Tidak Merespon

Tutup dan Buka Kembali Aplikasi: Coba tutup aplikasi dan buka kembali.

Lampiran 5. Sertifikat HKI

 REPUBLIK INDONESIA KEMENTERIAN HUKUM	
SURAT PENCATATAN CIPTAAN	
Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:	
Nomor dan tanggal permohonan	: EC002025078272, 30 Juni 2025
Pencipta	
Nama	: Dinda Ari Theanekawati, Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom. dkk
Alamat	: Jl. Sukosono, RT/RW 010/002, No. 28, Kelurahan Slerok, Tegal Timur, Kota Tegal, Jawa Tengah, 52125
Kewarganegaraan	: Indonesia
Pemegang Hak Cipta	
Nama	: Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama
Alamat	: Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana, Margadana, Kota Tegal, Jawa Tengah, 52142
Kewarganegaraan	: Indonesia
Jenis Ciptaan	: Program Komputer
Judul Ciptaan	: FFTalent: Sistem E-Recruitment dan Perangkingan Pemain Free Fire Menggunakan Metode KNN Berbasis Mobile
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	: 30 Juni 2025, di Kota Tegal
Jangka waktu perlindungan	: Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor Pencatatan	: 000918533
adalah benar berdasarkan keterangan yang diberikan oleh Pemohon. Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.	
	a.n. MENTERI HUKUM DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL u.b Direktur Hak Cipta dan Desain Industri  Agung Damarsasongko,SH.,MH. NIP. 196912261994031001
	Disclaimer: 1. Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan. 2. Surat Pencatatan ini telah disegel secara elektronik menggunakan segel elektronik yang diterbitkan oleh Balai Besar Sertifikasi Elektronik, Badan Siber dan Sandi Negara. 3. Surat Pencatatan ini dapat dibuktikan keasliannya dengan memindai kode QR pada dokumen ini dan informasi akan ditampilkan dalam browser.

Lampiran 6. Lembar Bimbingan



D IV TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

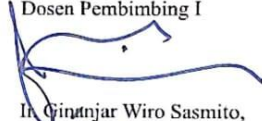
LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Dinda Ari Theanekawati
NIM : 21090071
No. Ponsel : 089619558721
Judul TA : Sistem E-Recruitment dan Perangkingan
Pemain Free Fire Berbasis Mobile dengan
Metode KNN dan SVM
Dosen Pembimbing I : Ir. Ginanjar Wiro Sasmito, M.Kom. IPM.,
ASEAN Eng

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1	23/25 4	Konsep	oke	b.
2.	26/25 5	prosedur	lanjut.	b.
3.	18/25 6	prosedur	lanjut	b.
4	26/25 5	prosedur	laport.	b.
5.			oke - lanjut Aki & draft artikel	b.

6.	07/07 ²⁵	Laporan	revisi	f.
67	08/07 ²⁵	Laporan	revisi	f.
8.	09/07 ²⁵	Laporan	Ok	f.
		All sing jrm		f.

Tegal,
Dosen Pembimbing I


Ir. Guntjar Wiro Sasmito,
M.Kom. IPM., ASEAN Eng
NIPY. 10.007.032



**D IV TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**

LEMBAR BIMBINGAN TUGAS AKHIR

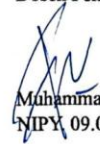
Nama : Dinda Ari Theanekawati
NIM : 21090071
No. Ponsel : 089619558721
Judul TA : Sistem E-Recruitment dan Perangkingan
Pemain Free Fire Berbasis Mobile dengan
Metode KNN dan SVM
Dosen Pembimbing II : Muhammad Fikri Hidayattullah, S.T., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	23 Mei 2024	Prototipe	- Integrasi ke sistem - Simulasi pada komputer selanjutnya	
2.	26 Juni 2025	Aplikasi	- Melaksanakan fitur seleksi - Daftar Hki	
3.	8 Juli	Laporan	Laporan	
4.	10 Juli	Laporan	Pengujian	

5.	10 juv	UML	Shinin	P ₂
6	12 juv	Splicing	Like Cates	P ₂
7	14 juv	Splicing	Splicing	P ₂
8	15 juv	Copied	Acc!	P ₂

--	--	--	--	--

Tegal,
Dosen Pembimbing II


Muhammad Fikri Hidayattullah, S.T., M.Kom.
NIPY 09.016.307