

K

MANUAL BOOK



RAIHAN HIDAYATULLAH DJUNAEDI

Daftar Isi

BLOCKCHAIN NETWORK.....	3
PERSIAPAN AWAL (FIRST SETUP)	3
MENGHASILKAN KUNCI NODE.....	3
MENYIAPKAN QBFT (QUORUM BYZANTINE FAULT TOLERANCE)	3
MEMULAI SEMUA NODE	3
KONFIRMASI SEMUA NODE DI JARINGAN	4
MENGHENTIKAN SEMUA NODE.....	4
MENAMBAHKAN VALIDATOR	4
MENGHAPUS VALIDATOR.....	4
MEMULAI NODE TERTENTU	5
MENGHENTIKAN NODE TERTENTU.....	5
KONFIRMASI NODE DI JARINGAN	5
MENGECEK JARINGAN DI QUORUM EXPLORER	5
SMART CONTRACT	7
DEPLOY SMART CONTRACT SKKM	7
PENGUJIAN SMART CONTRACT SKKM.....	7
STRUKTUR DIREKTORI PENTING	8
WEB APPLICATION.....	9
SETUP METAMASK	9
LANDING PAGE	10
SUPER ADMIN	12
STUDENT	14
HMJ.....	20
BEM	24

Blockchain Network

Persiapan Awal (First Setup)

Langkah pertama ini bertujuan untuk membersihkan direktori yang ada (kecuali folder `scripts` dan `quorum-explorer`) serta membuat konfigurasi jaringan blockchain default atau kustom.

- **Nama Skrip:** `0_firstSetup.sh`
- **Tujuan:** Membersihkan direktori dan membuat konfigurasi default atau kustom untuk blockchain.
- **Perintah untuk dijalankan:**

```
./scripts/0_firstSetup.sh
```
- **Opsi Konfigurasi:** Pilih antara menggunakan **konfigurasi default** atau memasukkan parameter kustom seperti chain ID, mekanisme konsensus, periode blok, dan jumlah node.

Menghasilkan Kunci Node

Skrip ini digunakan untuk menghasilkan kunci kriptografi yang diperlukan dan file konfigurasi blockchain seperti `genesis.json` untuk setiap node yang akan digunakan.

- **Nama Skrip:** `1_generateNodeKeys.sh`
- **Tujuan:** Menghasilkan kunci blockchain dan file konfigurasi untuk setiap node.
- **Perintah untuk dijalankan:**

```
./scripts/1_generateNodeKeys.sh
```
- **Hasil:** Folder bernama `networkFiles` yang berisi file `genesis.json` dan kunci untuk setiap node.

Menyiapkan QBFT (Quorum Byzantine Fault Tolerance)

Skrip ini menyiapkan direktori node dengan menyalin file `genesis.json` dan kunci node yang dihasilkan dari langkah sebelumnya.

- **Nama Skrip:** `2_setupQBFT.sh`
- **Tujuan:** Menyiapkan direktori node dengan file konfigurasi yang diperlukan.
- **Perintah untuk dijalankan:**

```
./scripts/2_setupQBFT.sh
```

Memulai Semua Node

Skrip ini menjalankan semua node dalam jaringan. Node pertama akan berfungsi sebagai bootnode, dan node berikutnya akan terhubung ke bootnode tersebut.

- **Nama Skrip:** `3_startAllNodes.sh`
- **Tujuan:** Memulai semua node dalam jaringan blockchain.

- **Perintah untuk dijalankan:**

```
./scripts/3_startAllNodes.sh
```

- **Hasil:** Log untuk setiap node akan dibuat di direktori logs.

Konfirmasi Semua Node di Jaringan

Skrip ini memastikan bahwa semua node berhasil bergabung ke dalam jaringan dan memverifikasi bahwa mekanisme konsensus berjalan dengan baik.

- **Nama Skrip:** 4_confirmAllNetwork.sh
- **Tujuan:** Memastikan bahwa semua node telah bergabung dalam jaringan dan konsensus berjalan.
- **Perintah untuk dijalankan:**

```
./scripts/4_confirmAllNetwork.sh
```

Menghentikan Semua Node

Skrip ini menghentikan semua node yang sedang berjalan dalam jaringan dengan menghentikan proses node.

- **Nama Skrip:** 5_stopAllNodes.sh
- **Tujuan:** Menghentikan seluruh node dalam jaringan.
- **Perintah untuk dijalankan:**

```
./scripts/5_stopAllNodes.sh
```

Menambahkan Validator

Skrip ini digunakan untuk menambahkan node baru sebagai validator ke jaringan. Node baru akan dimulai, diajukan sebagai validator, dan kemudian diterima oleh jaringan.

- **Nama Skrip:** 6_addValidator.sh
- **Tujuan:** Menambahkan validator baru ke dalam jaringan blockchain.
- **Perintah untuk dijalankan:**

```
./scripts/6_addValidator.sh
```

Menghapus Validator

Skrip ini digunakan untuk menghapus validator dari jaringan. Skrip ini membatalkan setiap suara yang belum diproses dan mengajukan penghapusan validator ke sebagian besar node yang ada.

- **Nama Skrip:** 7_removeValidator.sh
- **Tujuan:** Menghapus validator dari jaringan blockchain.
- **Perintah untuk dijalankan:**

```
./scripts/7_removeValidator.sh <alamat-validator>
```

Memulai Node Tertentu

Skrip ini digunakan untuk memulai node tertentu dalam jaringan blockchain.

- **Nama Skrip:** 8_startNodes.sh
- **Tujuan:** Memulai node tertentu dalam jaringan.
- **Perintah untuk dijalankan:**

```
./scripts/8_startNodes.sh <Node-X>
```

Menghentikan Node Tertentu

Skrip ini digunakan untuk menghentikan node tertentu dalam jaringan dengan menghentikan prosesnya.

- **Nama Skrip:** 9_stopNodes.sh
- **Tujuan:** Menghentikan node tertentu dalam jaringan.
- **Perintah untuk dijalankan:** ./scripts/9_stopNodes.sh <Node-X>

Konfirmasi Node di Jaringan

Skrip ini digunakan untuk memeriksa status node tertentu di jaringan blockchain, memastikan apakah node tersebut masih aktif dan berapa jumlah validator yang berpartisipasi di jaringan.

- **Nama Skrip:** 10_confirmNetwork.sh
- **Tujuan:** Mengonfirmasi apakah node tertentu di jaringan berfungsi dengan baik dan memverifikasi jumlah validator.
- **Perintah untuk dijalankan:**

```
./scripts/10_confirmNetwork.sh <Node-X>
```

Mengecek Jaringan di Quorum Explorer

Quorum Explorer secara otomatis dijalankan bersama dengan node-node blockchain melalui startAllNodes.sh. Untuk memverifikasi status jaringan, User dapat mengakses Quorum Explorer langsung dari browser.

Langkah-langkah Pengecekan:

1. **Buka Browser:** Buka browser favorit (seperti Google Chrome, Firefox, atau lainnya).
2. **Akses Quorum Explorer:** Pada address bar browser, ketik: <http://localhost:25000>
Lalu tekan **Enter**.
3. **Visualisasi Status Jaringan:** Setelah halaman Quorum Explorer terbuka, user akan melihat visualisasi dari jaringan blockchain, yang meliputi:
 - **Status Node:** User dapat melihat node mana yang aktif dan terhubung di jaringan.

- **Jumlah Validator:** Lihat jumlah validator yang sedang aktif di jaringan.
 - **Transaksi:** Monitor jumlah dan status transaksi yang terjadi di jaringan.
 - **Blok dan Konsensus:** User juga dapat melihat blok-blok yang telah terbentuk dan Consensus yang sedang berjalan.
4. **Pengecekan Validator dan Node:**
- Di antarmuka, Anda akan menemukan informasi yang lebih mendalam mengenai setiap **node** dan **validator**, termasuk jumlah validator yang aktif.
 - Pastikan semua **node validator** yang diharapkan telah bergabung dan berfungsi dengan benar.
5. **Memverifikasi Konsensus:** Di bagian Consensus, pastikan mekanisme Consensus (seperti QBFT) berfungsi dengan baik dengan status blok yang tercatat dan validator yang aktif.

Smart Contract

Deploy Smart Contract SKKM

Langkah-Langkah untuk Deploy Smart Contract

1. **Pastikan Semua Dependensi Terinstal:** Sebelum memulai proses deploy, pastikan semua dependensi sudah diinstal dengan menjalankan perintah berikut:

```
npm install
```

2. **Konfigurasi Jaringan di Hardhat:** Pastikan file **hardhat.config.js** telah dikonfigurasi dengan benar untuk jaringan **Besu**. Contoh konfigurasi:

```
module.exports = {
  solidity: "0.8.9",
  networks: {
    besu: {
      url: "http://127.0.0.1:8545", // URL jaringan Besu
      chainId: 1337,
      gasPrice: 0,
      accounts: [`0x${YOUR_PRIVATE_KEY}`] // Gantilah dengan
      private key deployer
    },
  },
};
```

3. **Deploy Smart Contract:** Setelah jaringan **Besu** aktif, deploy smart contract **SKKM** menggunakan perintah berikut:

```
npx hardhat run scripts/deploy.js --network besu
```

- o Perintah ini akan melakukan deploy smart contract **SKKM** di jaringan **Besu** sesuai dengan konfigurasi yang ada di file **deploy.js**.
- o Setelah berhasil di-deploy, alamat kontrak akan disimpan di file **contractaddress.json**.

Pengujian Smart Contract SKKM

Setelah smart contract di-deploy, pengujian diperlukan untuk memastikan semua fungsi berjalan sesuai dengan yang diharapkan.

Langkah Menjalankan Pengujian Smart Contract

1. **Menjalankan Tes Blockchain:** Untuk menjalankan pengujian khusus pada blockchain, gunakan perintah berikut:

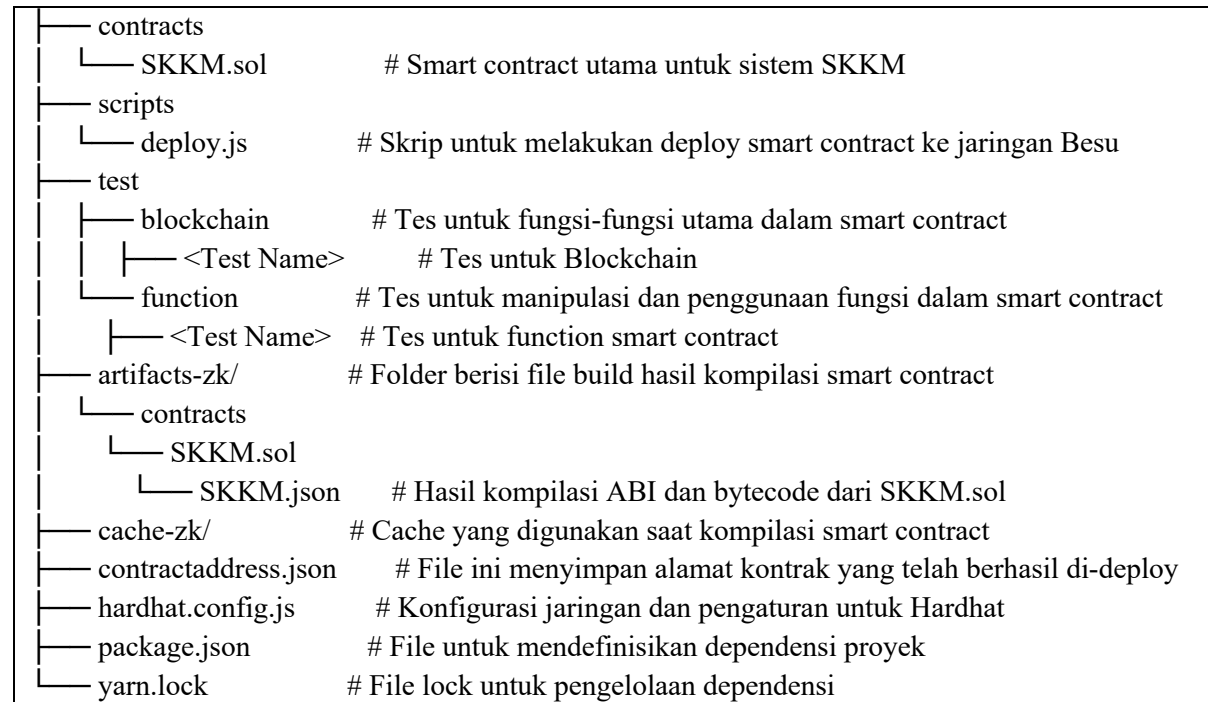
```
npx hardhat test test/blockchain/<Nama Test> --network besu
```

2. **Menjalankan Tes Function:** Untuk menjalankan pengujian khusus pada fungsi tertentu, jalankan:

```
npx hardhat test test/function/<Nama Test> --network besu
```

Struktur Direktori Penting

Setelah proses deploy dan pengujian smart contract selesai, penting untuk memahami struktur folder dan file yang digunakan dalam proyek ini. Struktur direktori berikut menunjukkan elemen-elemen penting dari proyek **Blockchain Academic Record** yang terkait dengan pengembangan dan pengujian smart contract **SKKM**.



Web Application

Setup Metamask

The screenshot shows the Metamask 'Add a network' interface. On the left, under 'Test networks', 'Ledger Kuliah' is selected and highlighted with a red box. On the right, the 'Add a network' form is displayed with the following fields: 'Network name' (Ledger Kuliah), 'New RPC URL' (http://localhost:8545), 'Chain ID' (1337), 'Currency symbol' (ETH), and 'Block explorer URL (Optional)' (http://localhost:25000/explorer/nodes). The 'Save' button is highlighted with a red box.

Untuk **setup Metamask** pada web application, berikut adalah langkah-langkahnya yang bisa diikuti:

1. Buka Ekstensi Metamask

- Klik ikon Metamask di sudut kanan atas browser untuk membuka ekstensi Metamask.

2. Tambahkan Jaringan Baru

- Pada tampilan utama Metamask, klik pada bagian atas yang menunjukkan jaringan saat ini (misalnya, "Ethereum Mainnet").
- Gulir ke bawah dan pilih "Add a Network" atau "Custom RPC".

3. Isi Informasi Jaringan

- **Network Name:** Ledger Kuliah (atau sesuai dengan nama jaringan yang ingin Anda gunakan).
- **New RPC URL:** Masukkan URL RPC dari node Besu, misalnya http://localhost:8545.
- **Chain ID:** Isi dengan 1337 (untuk jaringan lokal).
- **Currency Symbol:** Masukkan ETH sebagai simbol mata uang.
- **Block Explorer URL (Optional):** Masukkan URL block explorer jika tersedia, misalnya http://localhost:25000/explore/nodes.

4. Simpan Konfigurasi

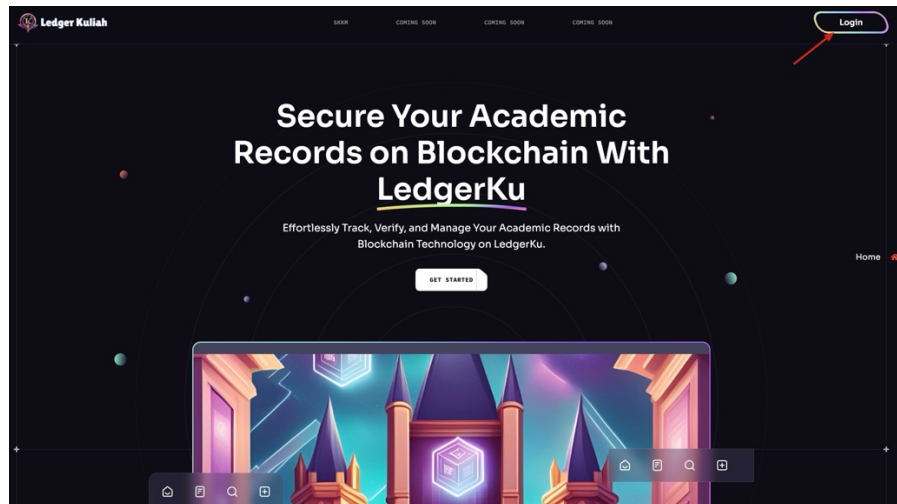
- Setelah mengisi semua informasi, klik tombol **Save** untuk menyimpan jaringan yang baru Anda tambahkan.

5. Verifikasi Jaringan

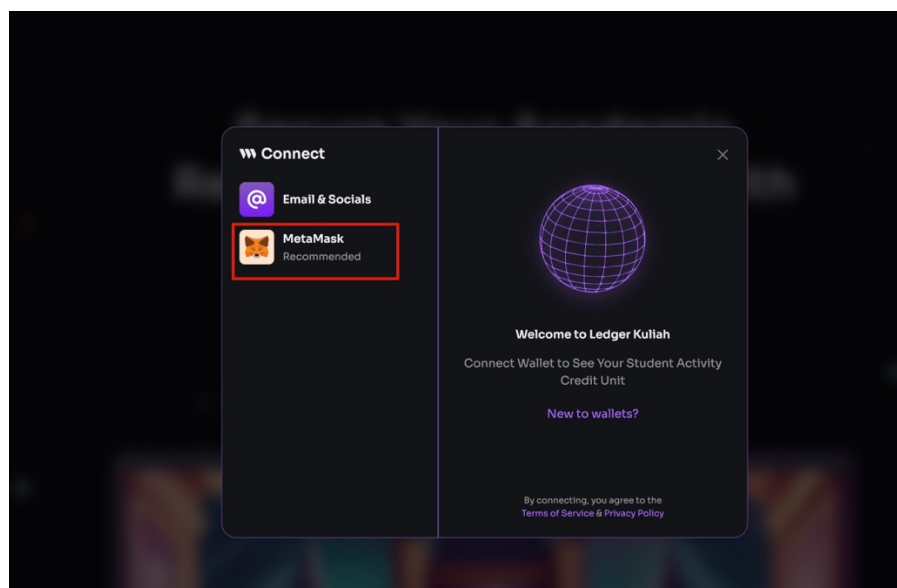
- Setelah jaringan berhasil ditambahkan, pastikan Metamask terhubung ke jaringan yang baru saja Anda tambahkan, dengan melihat nama jaringan di bagian atas (misalnya **Ledger Kuliah**).

Landing Page

1. Login ke Aplikasi Web LedgerKu
 - a. Buka aplikasi web LedgerKu.
 - b. Klik tombol Login di pojok kanan atas halaman utama untuk memulai proses login ke aplikasi.

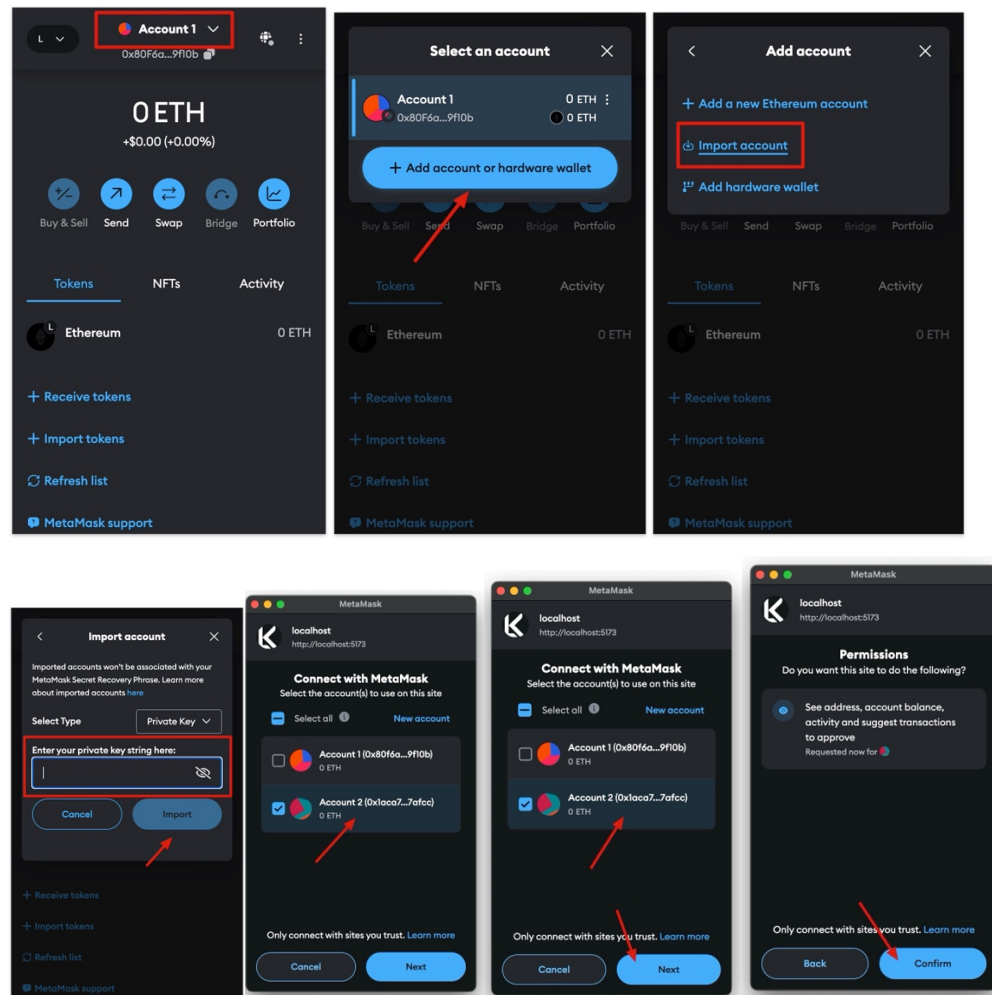


2. Pilih Opsi Koneksi MetaMask
 - a. Setelah mengklik Login, akan muncul layar pilihan koneksi dompet.

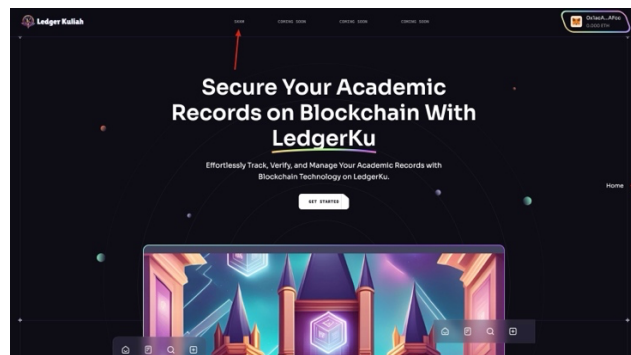


- b. Pilih MetaMask dari pilihan yang ada untuk menghubungkan aplikasi dengan dompet MetaMask. Jika perlu menggunakan akun lain atau menambahkan akun baru, klik opsi **"Add account or hardware wallet"** di MetaMask. Untuk mengimpor akun baru, pilih **"Import account"** dan masukkan kunci privat dari akun tersebut. Setelah akun yang diinginkan sudah dipilih atau ditambahkan, kembali ke proses login di aplikasi LedgerKu. Pada jendela pop-up MetaMask yang muncul, pilih akun yang akan dihubungkan dengan aplikasi. Klik tombol **"Next"** untuk melanjutkan proses koneksi.

MetaMask akan meminta konfirmasi untuk menghubungkan akun dengan aplikasi LedgerKu. Klik tombol "**Confirm**" untuk menyelesaikan proses koneksi.



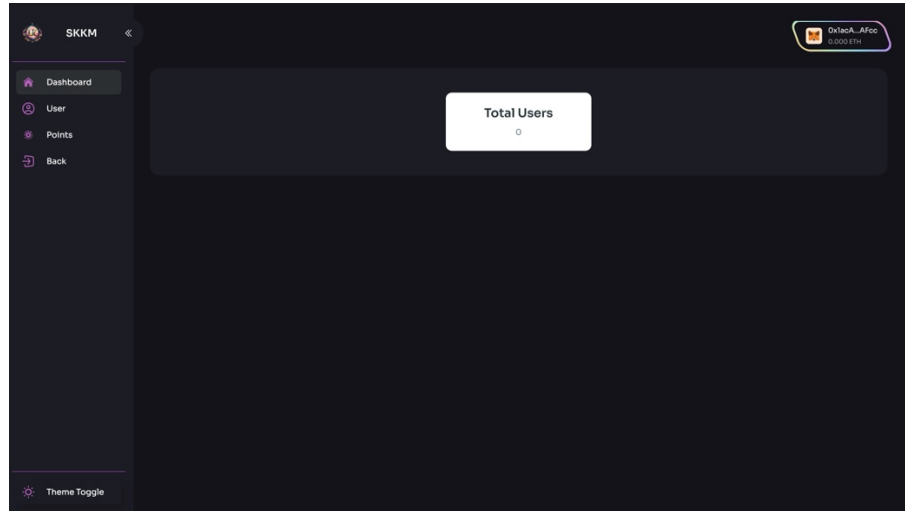
- Setelah berhasil login dengan akun MetaMask, Anda akan diarahkan ke halaman utama seperti yang ditunjukkan pada gambar. Pada halaman ini, Anda dapat melihat berbagai opsi menu di bagian atas, seperti **SKKM** dan beberapa menu lain yang bertuliskan **Coming Soon**. Klik pada menu **SKKM** untuk mengakses fitur-fitur terkait pencatatan aktivitas akademik menggunakan blockchain. Alamat wallet MetaMask yang terhubung akan ditampilkan di pojok kanan atas layar bersama dengan saldo ETH yang tersedia.



Super Admin

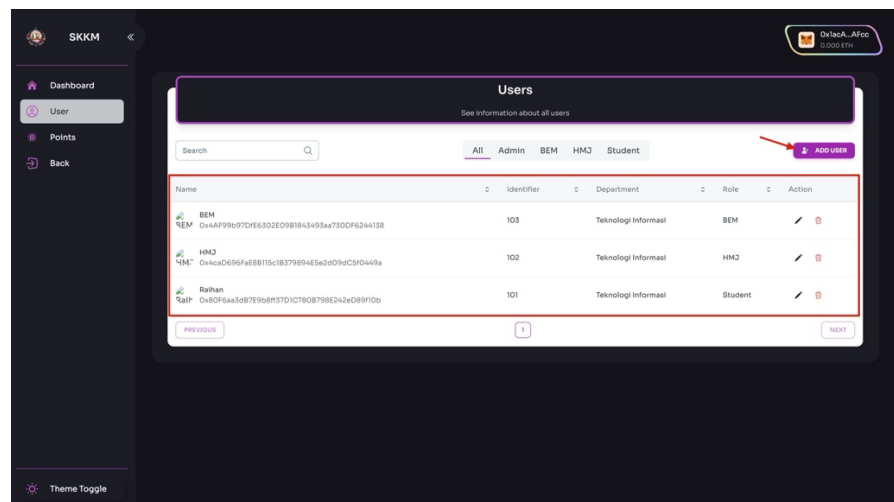
1. Dashboard

- Gambar di bawah menunjukkan tampilan awal setelah super admin login. Di halaman ini, super admin dapat melihat jumlah total pengguna yang ada dalam sistem.



2. Melihat Data Pengguna

- Setelah memilih menu "User", super admin akan diarahkan ke halaman daftar pengguna. Di sini, semua pengguna yang terdaftar dalam sistem akan muncul beserta informasi lengkapnya seperti *name*, *identifier*, *department*, dan *role*. Super admin juga dapat menambah, mengedit, atau menghapus pengguna.



3. Menambah Pengguna Baru

- Untuk menambah pengguna baru, klik tombol *Add User* yang berada di pojok kanan atas halaman *User*.
- Setelah itu, isi informasi pengguna yang baru seperti alamat, nama, identifier, departemen, dan peran (*role*) mereka. Klik *Add User* setelah semua informasi diisi dengan benar.

The 'Add User' form is displayed within the SKKM application interface. It features a sidebar with navigation options: Dashboard, User, Points, and Back. The form itself has a title 'Add User' and a subtitle 'Fill out the form below to add a new user'. The fields are: Address, Name, Identifier, Department, and Role (with a dropdown menu showing 'Student'). A 'Back' button is located at the bottom left of the form, and an 'Add User' button is at the bottom right. Red arrows highlight each input field and the 'Add User' button.

4. Mengedit Data Pengguna

- Super admin dapat mengedit data pengguna dengan mengklik ikon pensil di sebelah pengguna yang ingin diubah. Pada halaman edit, super admin dapat memperbarui informasi pengguna yang sudah ada.

The 'Update User' form is displayed within the SKKM application interface. It features a sidebar with navigation options: Dashboard, User, Points, and Back. The form has a title 'Update User' and a subtitle 'Fill out the form below to update the user'. The fields are: Address, Name, Identifier, Department, and Role (with a dropdown menu showing 'Student'). A 'Back' button is located at the bottom left of the form, and an 'Update User' button is at the bottom right. Red arrows highlight each input field and the 'Update User' button.

5. Menghapus Pengguna

- Jika diperlukan, super admin bisa menghapus pengguna dengan menekan ikon tong sampah di daftar pengguna. Sistem akan meminta konfirmasi sebelum pengguna benar-benar dihapus dari sistem.

A confirmation dialog box is shown with the text: 'Are you sure you want to delete this user?'. Below this, it says: 'This action cannot be undone. The user Raihan will be permanently removed from the system.' At the bottom right, there are two buttons: 'CANCEL' and 'CONFIRM'. Red arrows point to the 'CONFIRM' button.

6. Mengatur Poin Minimal

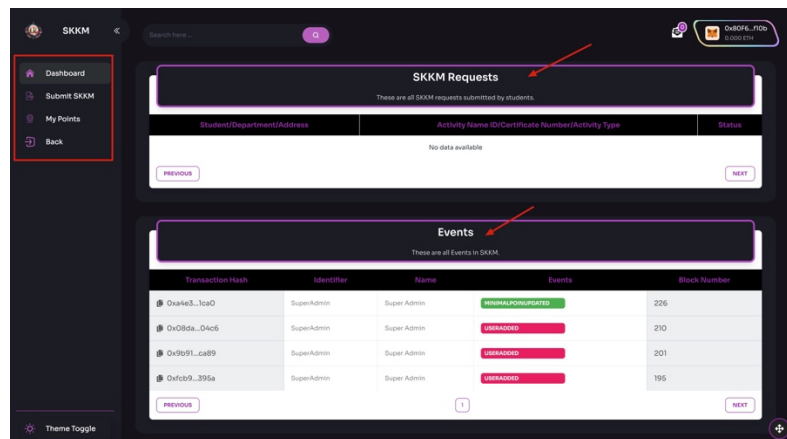
- Super admin juga memiliki akses untuk menentukan poin minimal yang diperlukan untuk berbagai aktivitas di sistem. Pada halaman "Points", super admin dapat memasukkan angka poin minimal yang diinginkan dan menekan tombol *Set Minimal Poin* untuk menyimpan perubahan.

The 'Minimal Poin' form is displayed within the SKKM application interface. It features a sidebar with navigation options: Dashboard, User, Points, and Back. The form has a title 'Minimal Poin' and a subtitle 'Current Minimal Poin : 0.0'. There is an input field for 'New Minimal Poin' with the value '3.5'. A 'SET MINIMAL POIN' button is located at the bottom right. Red arrows point to the input field and the button.

Student

1. Dashboard Mahasiswa

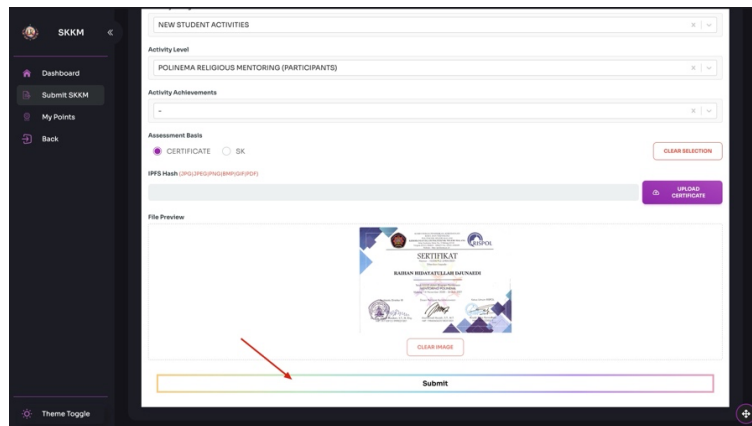
- Gambar di bawah menunjukkan tampilan **Dashboard** setelah mahasiswa login ke dalam sistem. Pada halaman ini, mahasiswa dapat melihat berbagai permintaan SKKM yang telah diajukan oleh mahasiswa lainnya serta melihat **Events** terkait aktivitas yang sudah diproses oleh sistem.



2. Mengajukan SKKM (Submit Activity)

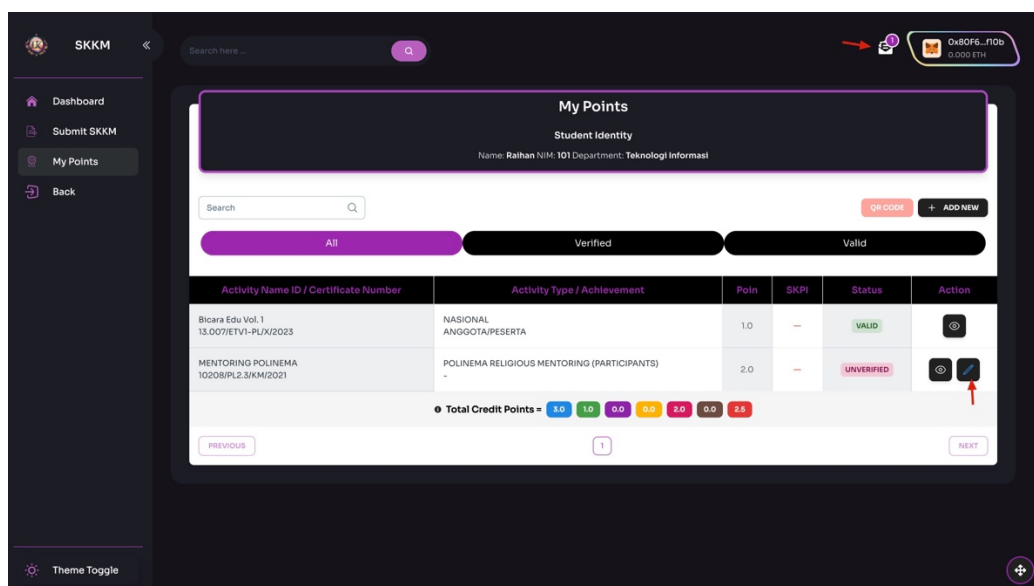
- Setelah memilih menu **Submit SKKM**, mahasiswa akan diarahkan ke halaman pengajuan aktivitas baru. Pada halaman ini, mahasiswa harus mengisi informasi aktivitas yang akan diajukan. Informasi yang perlu diisi meliputi **Nama Aktivitas (Bahasa Indonesia)**, **Nama Aktivitas (English)**, **Nomor Sertifikat**, **Kategori Aktivitas**, **Level Aktivitas**, dan **Pencapaian Aktivitas**. Sertifikat kegiatan juga harus diunggah melalui opsi **Upload Certificate**. Setelah semua informasi diisi dengan benar, mahasiswa dapat menekan tombol **Submit** untuk mengajukan aktivitasnya ke sistem.

The screenshot shows the 'Submit Activity Credit' form. It contains several input fields and dropdown menus, each with a red arrow pointing to it: 'Activity Name (BHS INDONESIA)', 'Activity Name (ENGLISH)', 'Certificate Number', 'Activity Categories', 'Activity Level', 'Activity Achievements', and 'Assessment Basis'. At the bottom right, there is a purple button labeled 'UPLOAD CERTIFICATE' with a red arrow pointing to it. The sidebar on the left shows the 'Submit SKKM' menu highlighted.



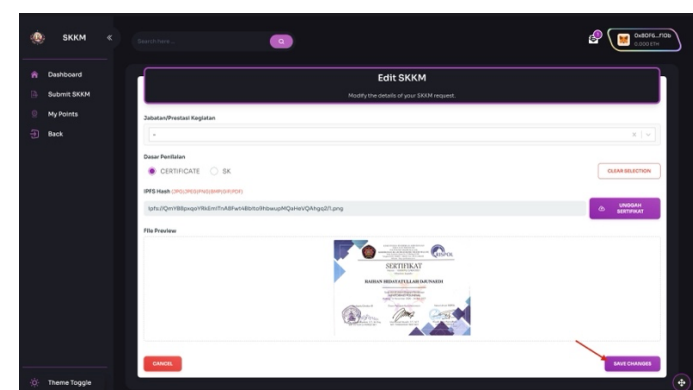
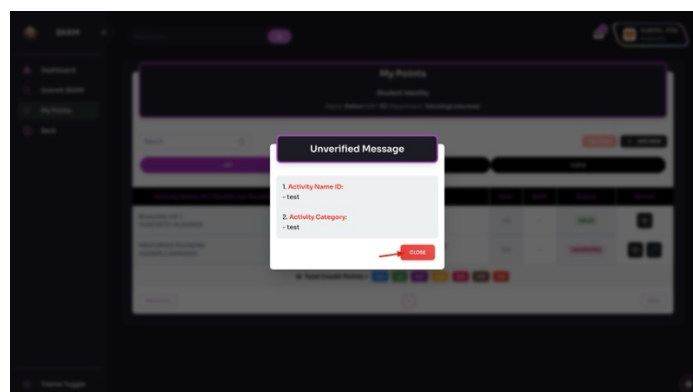
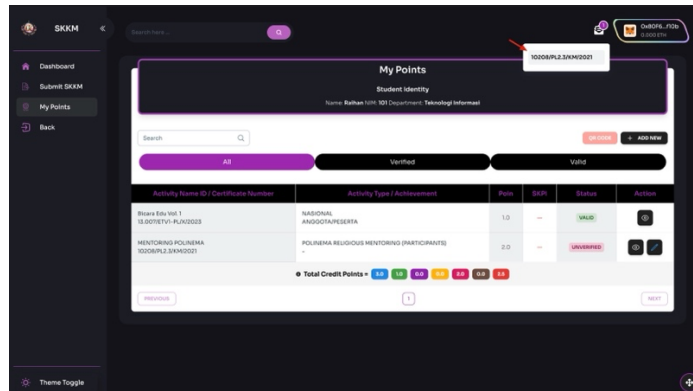
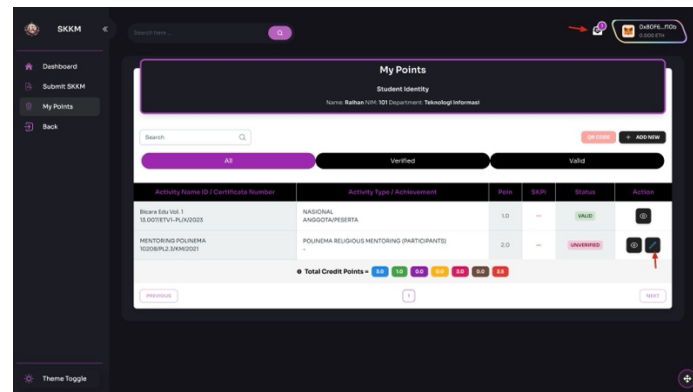
3. Melihat dan Mengelola Poin Aktivitas (My Points)

- Mahasiswa dapat mengakses menu **My Points** untuk melihat daftar aktivitas yang sudah diajukan beserta status verifikasi. Pada tabel yang tersedia, terdapat informasi mengenai **Nama Aktivitas**, **Tipe Aktivitas**, **Poin SKKM**, dan status verifikasi. Jika diperlukan, mahasiswa juga dapat mengedit data aktivitas yang belum diverifikasi dengan menekan ikon pensil pada kolom **Action**.



4. Mengedit Pengajuan Aktivitas

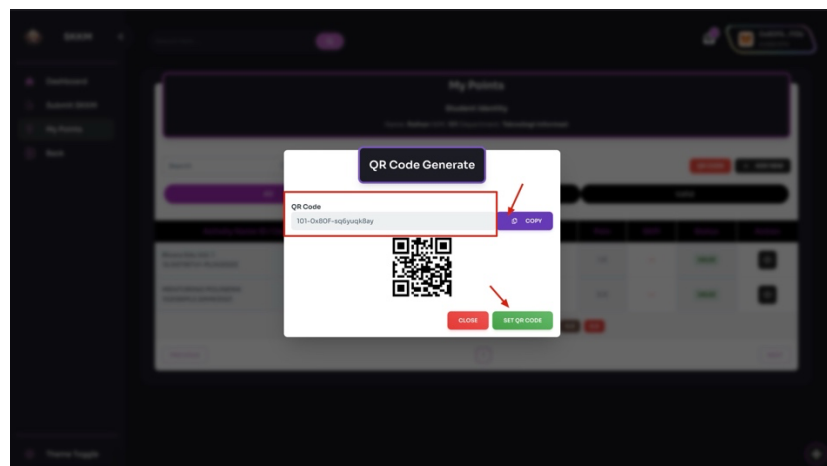
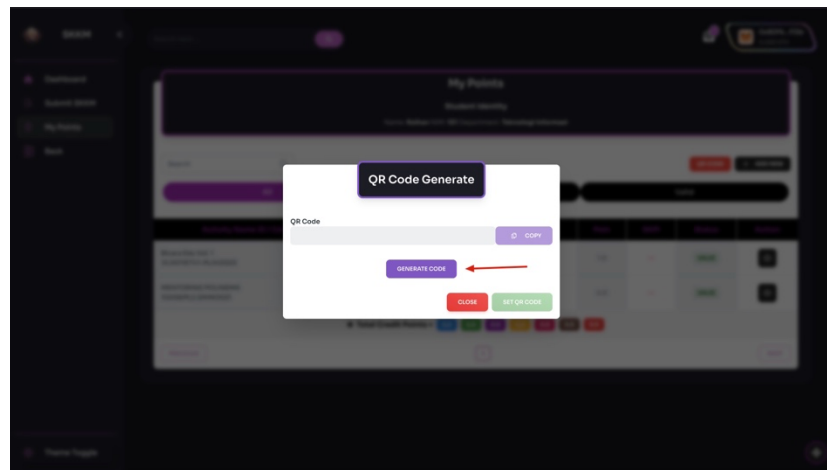
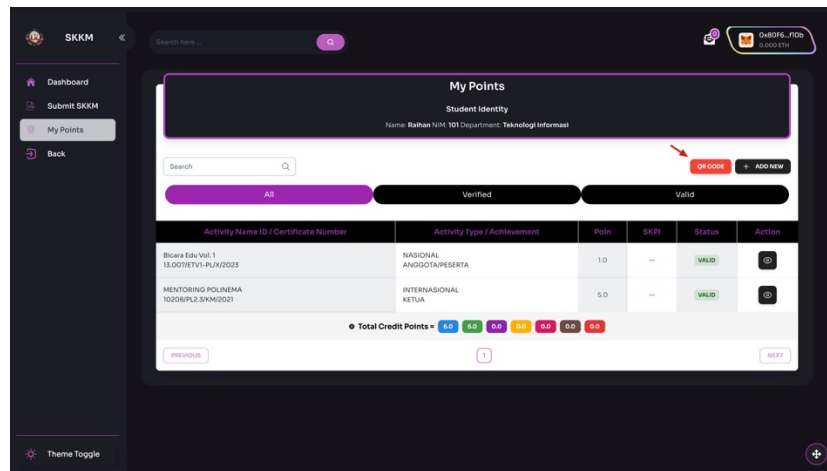
- Untuk mengedit aktivitas yang sudah diajukan, mahasiswa dapat masuk ke menu **My Points** dan mengklik ikon pensil di sebelah aktivitas yang ingin diubah. Pada halaman edit, mahasiswa dapat memperbarui informasi aktivitas seperti **Dasar Penilaian** dan **Unggahan Sertifikat**. Setelah melakukan perubahan, tekan tombol **Save Changes** untuk menyimpan perubahan.



5. Membuat Kode QR untuk Verifikasi (Generate QR Code)

- Mahasiswa juga memiliki opsi untuk menghasilkan **Kode QR** untuk memudahkan proses verifikasi aktivitas oleh SEB (Student Executive Board). Kode QR ini dapat dibuat dengan menekan tombol **QR Code** yang berada di bagian kanan halaman **My Points**. Setelah kode

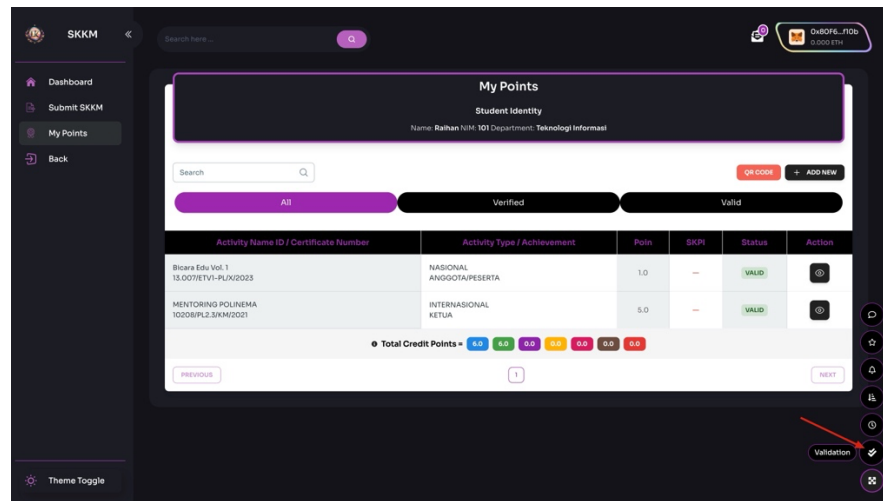
dihasilkan, mahasiswa bisa menyalin atau menetapkan kode tersebut dengan menekan tombol **Set QR Code**.



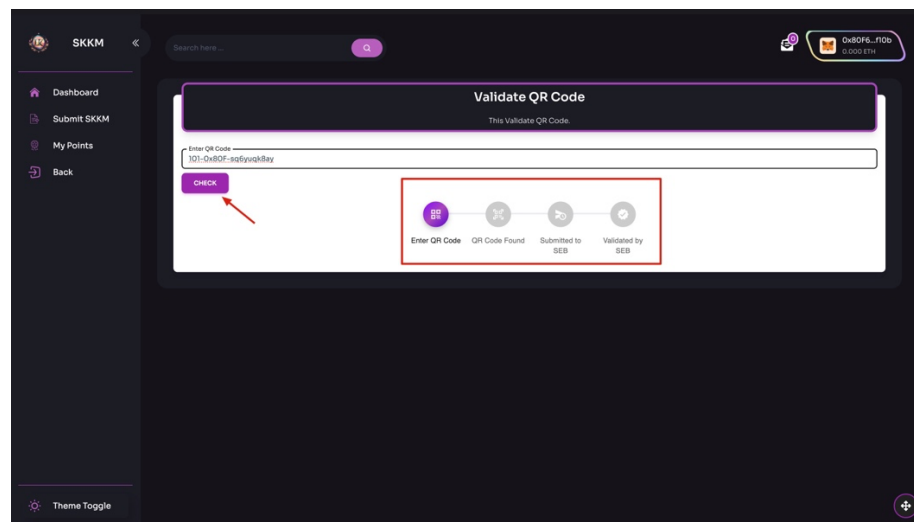
6. Memvalidasi Kode QR (Validate QR Code)

- Kode QR yang telah dihasilkan dapat digunakan oleh pihak SEB untuk memverifikasi keaslian aktivitas yang diajukan oleh mahasiswa. Proses validasi ini dilakukan dengan memasukkan kode QR pada halaman **Validate QR Code**. Setelah kode dimasukkan, sistem akan

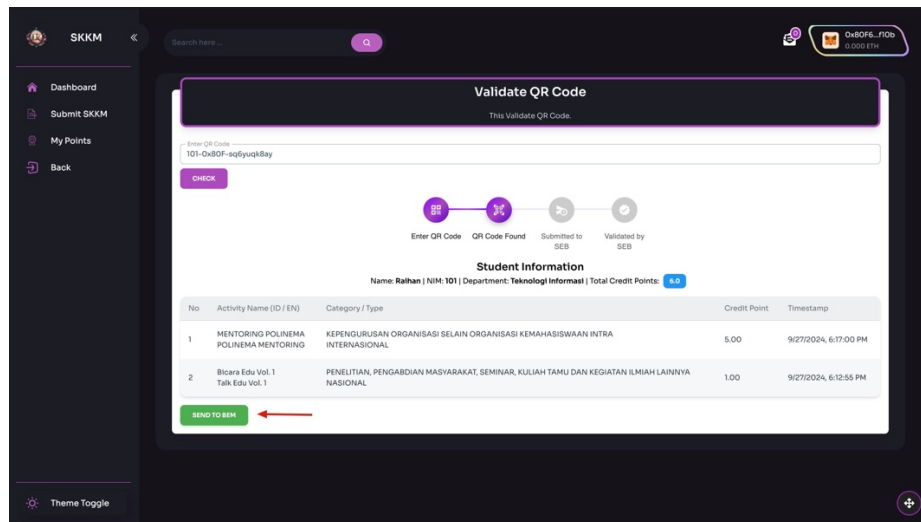
menampilkan detail aktivitas mahasiswa yang diverifikasi dan pihak SEB dapat melanjutkan proses verifikasi.



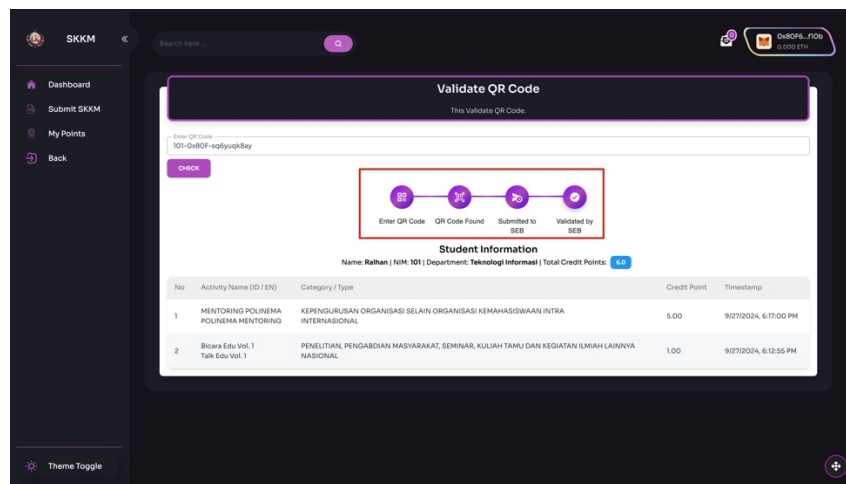
- Pada halaman validasi QR Code, pengguna harus memasukkan kode QR yang telah dihasilkan sebelumnya. Gambar di atas menunjukkan kolom untuk memasukkan kode QR dan tombol "Check" untuk memulai proses validasi.



- Setelah kode QR dimasukkan dan tombol "Check" ditekan, sistem akan menampilkan status validasi. Gambar menunjukkan langkah-langkah validasi yang meliputi:
 - Enter QR Code: Memasukkan kode QR.
 - QR Code Found: Sistem berhasil menemukan kode QR.
 - Submitted to SEB: Aktivitas telah diajukan ke Student Executive Board (SEB).
 - Validated by SEB: Aktivitas berhasil divalidasi oleh SEB.



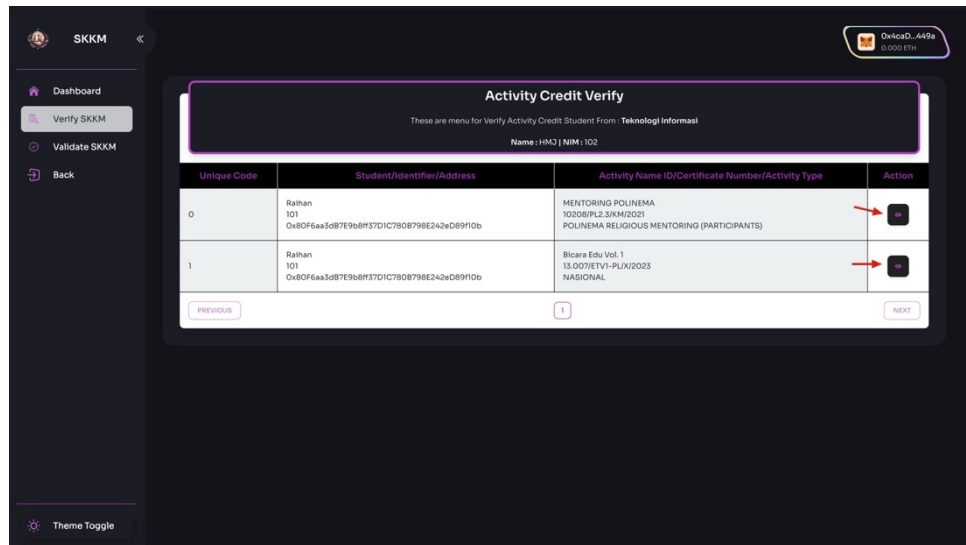
- Setelah QR Code ditemukan dan diverifikasi oleh SEB, informasi aktivitas mahasiswa akan muncul di halaman. Pengguna dapat melihat aktivitas, kategori, poin kredit, dan timestamp dari aktivitas tersebut. Untuk menyelesaikan proses, pengguna dapat mengirimkan aktivitas ke BEM (Badan Eksekutif Mahasiswa) dengan menekan tombol "Send to BEM".



HMJ

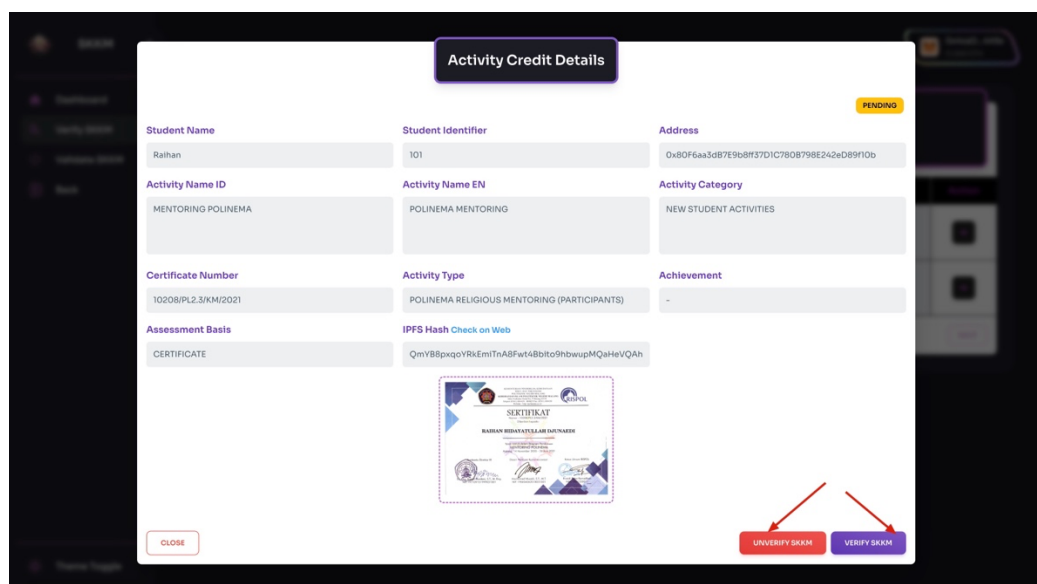
1. Dashboard HMJ

- Gambar di bawah menunjukkan tampilan Dashboard setelah pengguna HMJ login. Di halaman ini, pengguna HMJ dapat melihat daftar aktivitas yang diajukan oleh mahasiswa, beserta informasi lengkapnya seperti Nama Mahasiswa, NIM, Nama Aktivitas, dan Jenis Sertifikat.



2. Memverifikasi atau Tidak Memverifikasi Aktivitas

- Pada halaman detail aktivitas, terdapat dua tombol utama: **Verify SKKM** dan **Unverify SKKM**. Jika aktivitas sesuai dengan kriteria, pengguna HMJ dapat menekan tombol **Verify SKKM** untuk memvalidasi aktivitas. Jika ada kesalahan atau aktivitas perlu revisi, pengguna HMJ dapat memilih tombol **Unverify SKKM** dan memberikan catatan perbaikan.



3. Unverify Aktivitas

- Jika aktivitas tidak sesuai atau perlu revisi, pengguna dapat memilih tombol **Unverify SKKM**. Setelah menekan tombol ini, akan muncul opsi untuk memilih bagian mana yang harus

diperbaiki, seperti Nama Aktivitas, Kategori Aktivitas, dan Dasar Penilaian. Pengguna juga dapat menambahkan pesan penjelasan sebelum mengirimkan permintaan revisi ke mahasiswa.

Activity Credit Details

Student Name: Raihan, Student Identifier: 101, Address: 0x80F6aa3d879b8f3d1c7b08798e242d69f0b

Activity Name ID: MENTORING POLINEMA, Activity Name EN: POLINEMA MENTORING, Activity Category: NEW STUDENT ACTIVITIES

Certificate Number: 10208PL2.3/KM/2021, Activity Type: POLINEMA RELIGIOUS MENTORING (PARTICIPANTS), Achievement: -

Assessment Basis: CERTIFICATE, IPFS Hash Check on Web: QmY8BpgoY9kEmTnABFw4Boto9bwpMQaHeVQAn

Buttons: CLOSE, UNVERIFY SKKM, VERIFY SKKM

Unverify SKKM

Select the fields that need to be revised **before** enter a message.

☒ Activity Name ID, ☐ Activity Name EN, ☐ Certificate Number, ☐ Achievement, ☐ IPFS Hash

☐ Activity Category, ☐ Activity Type, ☐ Assessment Basis

1: Activity Name ID: -test
2: Activity Category: -test

Buttons: CLOSE, SUBMIT

4. Melihat Pesan Revisi

- Setelah aktivitas direvisi oleh mahasiswa, pengguna HMJ dapat melihat pesan revisi yang telah dikirimkan oleh sistem. Dengan menekan tombol **See Message**, pengguna dapat memeriksa pesan perbaikan yang telah diberikan dan melakukan verifikasi ulang setelah aktivitas diperbaiki.

Activity Credit Details

Student Name: Raihan, Student Identifier: 101, Address: 0x80F6aa3d879b8f3d1c7b08798e242d69f0b

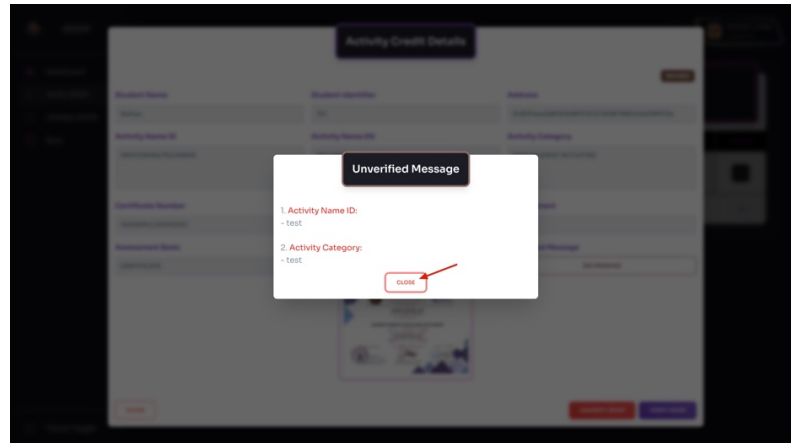
Activity Name ID: MENTORING POLINEMA, Activity Name EN: POLINEMA MENTORING, Activity Category: NEW STUDENT ACTIVITIES

Certificate Number: 10208PL2.3/KM/2021, Activity Type: POLINEMA RELIGIOUS MENTORING (PARTICIPANTS), Achievement: -

Assessment Basis: CERTIFICATE, IPFS Hash Check on Web: QmY8BpgoY9kEmTnABFw4Boto9bwpMQaHeVQAn

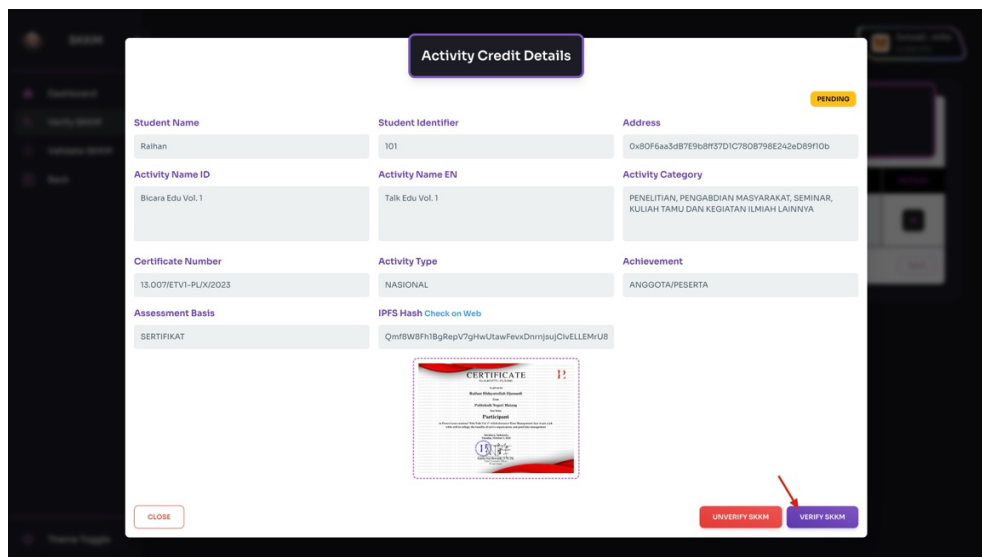
Unverified Message: SEE MESSAGE

Buttons: CLOSE, UNVERIFY SKKM, VERIFY SKKM



5. Memverifikasi Aktivitas Mahasiswa

- Setelah memilih menu **Verify SKKM**, pengguna HMJ akan diarahkan ke halaman verifikasi aktivitas. Pengguna dapat melihat daftar aktivitas yang siap diverifikasi, dengan rincian seperti Kode Unik, Nama Mahasiswa, NIM, Nama Aktivitas, dan Tipe Aktivitas. Pengguna dapat menekan tombol **Lihat** untuk melihat detail aktivitas.



6. Memvalidasi Aktivitas Mahasiswa

- Pada halaman **SKKM Details**, setelah pengguna HMJ memeriksa detail aktivitas mahasiswa seperti:
 - **Kategori Aktivitas**
 - **Basis Penilaian (Sertifikat)**
 - **Poin Kredit Aktivitas**
- Pengguna HMJ harus memeriksa semua informasi yang diberikan. Setelah itu, pengguna dapat menekan tombol **Validate SKKM** yang terletak di bagian bawah halaman untuk memvalidasi aktivitas mahasiswa.
- Proses validasi ini akan memastikan bahwa aktivitas yang diajukan telah diverifikasi dan diakui oleh sistem SKKM.

SKKM Details

Student Name Raihan	Student Identifier 101	Address 0x80F6aa3d87E9b8ff37D1C780B798E242eD89f10b
Activity Name ID Bicara Edu Vol.1	Activity Name EN Talk Edu Vol.1	Certificate Number 13.007/ETVI-PLX/2023
Activity Category PENELITIAN, PENGABDIAN MASYARAK... x	Activity Type NASIONAL x	Achievement ANGGOTA/PESERTA x
Assessment Basis SERTIFIKAT x	Credit Poin 1	IPFS Hash Check on Web Qmrf8W8Fh1BgRepV7gHwUtawFevxDnmjsujCivELLEMfU8



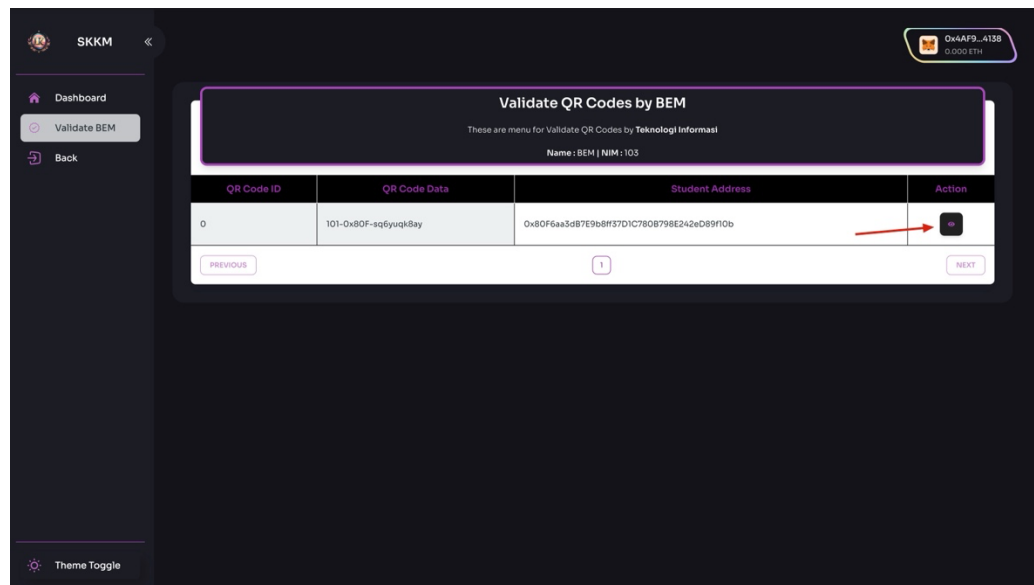
CLOSE

VALIDATE SKKM

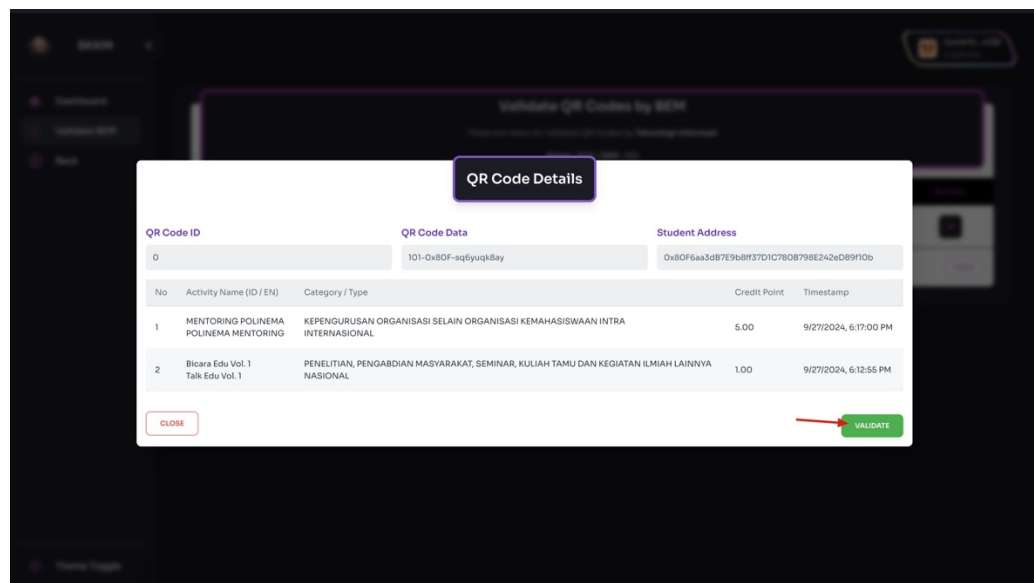
BEM

1. Memverifikasi QR Code Aktivitas oleh BEM

- Setelah memilih menu **Validate BEM**, pengguna BEM akan diarahkan ke halaman verifikasi QR Code aktivitas. Pada halaman ini, BEM dapat melihat daftar QR Code yang siap diverifikasi, dengan rincian seperti **QR Code ID**, **Data QR Code**, dan **Alamat Mahasiswa**. Pengguna BEM dapat menekan tombol **Lihat** (ikon mata) untuk melihat detail aktivitas.



- Pada halaman detail QR Code, BEM dapat melihat informasi lengkap terkait aktivitas mahasiswa seperti **Nama Aktivitas**, **Kategori Aktivitas**, **Poin SKKM**, dan **Waktu Pengajaran**. Untuk memvalidasi aktivitas, pengguna BEM dapat menekan tombol **VALIDATE** yang berada di bagian bawah halaman.



Setiap langkah kecil yang kita ambil membawa kita lebih dekat pada pencapaian besar. Jangan takut untuk mencoba, belajar, dan terus berkembang. Dalam setiap proses, ada pelajaran berharga yang akan membentuk masa depan kita

- RHD -