|  |
| --- |
| // 1. Perhatikan code berikut! |
|  | /\* |
|  | const Card = () => { |
|  | const mapProducts = () => { |
|  | products.map((item, idx) => { |
|  | <div key={idx}>{item}</div> |
|  | }) |
|  | } |
|  |  |
|  | return( |
|  | <> |
|  | {mapProducts()} |
|  | </> |
|  | ) |
|  | } |
|  | } |
|  | export default Card |
|  | \*/ |
|  | // Apa yang akan terjadi jika code ini dijalankan? |
|  | // a. Component akan tetap kosong, karena parameter di dalam map tidak menggunakan |
|  | // syntax value dan index |
|  | // b. Component akan me-render setiap data yang ada di dalam array products |
|  | // x Component akan tetap kosong karena mapProducts tidak me-return hasil mapping |
|  | // yang dilakukan |
|  | // d. Component akan tetap kosong, karena React tidak menerima array sebagai hasil |
|  | // return sebuah component |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 2. Apa yang akan menjadi warna akhir dari text paragraf di bawah ini? |
|  | /\* |
|  | <style> |
|  | .text-paragraf{ |
|  | color: 'red'; |
|  | } |
|  |  |
|  | #text-paragraf{ |
|  | color: 'blue'; |
|  | } |
|  | </style> |
|  | <body> |
|  | <p style="background-color: green"> |
|  | Lorem ipsum |
|  | </p> |
|  | </body> |
|  | \*/ |
|  | // x Black |
|  | // b. Red |
|  | // c. Blue |
|  | // d. Green |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 3. Berikut ini command yang tepat untuk membuat sebuah project react, yaitu? |
|  | // x npx create-react-app my-new-app |
|  | // b. npm create-react-app my-first-app |
|  | // c. npx create-react-app Myfirstreactapp |
|  | // d. Semua jawaban benar! |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 4. Perhatikan command berikut: npx create-react-app myProjectApp |
|  | // Apakah maksud dari command "myProjectApp"? |
|  | // a. Tipe dari aplikasi yang akan kita buat |
|  | // b. Referensi dari project lain untuk aplikasi yang akan kita buat |
|  | // c. Sebuah package manager yang akan disimpan ke dalam node modules project |
|  | // React kita |
|  | // x Memberi nama project untuk CRA (create-react-app) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 5. Element manakah yang memiliki default display inline? |
|  | // a. div |
|  | // b. h6 |
|  | // x em |
|  | // d. hr |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 6. Bagaimanakah cara meng-import file css dengan benar? |
|  | // x <link rel="stylesheet" type="text/css" href="externalStylesheet.css"> |
|  | // b. <link rel="stylesheet" type="text/css" src="externalStylesheet.css"> |
|  | // c. <stylesheet src="externalStylesheet.css"></stylesheet> |
|  | // d. <stylesheet>externalStylesheet.css</stylesheet> |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 7. Apa yang akan terjadi apabila kita menyimpan data ke dalam variable, |
|  | // kemudian melakukan manipulasi pada variable tersebut? |
|  | // a. Data didalam variable tidak akan berubah karena harus menggunakan state untuk menyimpan data |
|  | // x Data didalam variable berubah tetapi component tidak akan auto re-render |
|  | // c. Data didalam variable berubah dan component akan auto re-render |
|  | // d. Error, karena di React JS sudah memiliki fitur state untuk menyimpan data |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 8. Berikut ini yang merupakan selector di dalam CSS, kecuali? |
|  | // a. Tag selector |
|  | // b. Class selector |
|  | // c. Id selector |
|  | // x Element selector |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 9. Perhatikan code berikut! |
|  | /\* |
|  | const greeting = isLoggedIn? <Lists /> : null // Ada case lain |
|  | \*/ |
|  | // Kapan component Lists akan tampil? |
|  | // x Ketika isLoggedIn bernilai true |
|  | // b. Ketika user melakukan login |
|  | // c. Ketika function Lists dipanggil |
|  | // d. Tidak akan pernah tampil |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 10. Perhatikan code berikut! |
|  | /\* |
|  | <div class='box'> |
|  | <div> |
|  | JCWD-2302 Nih Bos! |
|  | </div> |
|  | <div> |
|  | Siap, Siap, Siap! |
|  | <div> |
|  | Nanaonan. |
|  | </div> |
|  | </div> |
|  | </div> |
|  | \*/ |
|  | // Selector yang tepat untuk men-styling "Siap, Siap, Siap!" tanpa mempengaruhi kata "Nanaonan"? |
|  | // x .box > div |
|  | // b. .box div |
|  | // c. .box |
|  | // d. div |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 11. Hooks apa yang digunakan untuk mendeklarasikan suatu local state |
|  | // dan dapat memperbarui nilai state tersebut? |
|  | // a. state |
|  | // x useState |
|  | // c. unitedState |
|  | // d. useContext |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 12. Berikut ini macam-macam method/function di dalam React Hooks, kecuali? |
|  | // a. useState //State didalam hooks |
|  | // b. useEffect //Lifecycle method didalam hooks |
|  | // x useLocation //Punyanya react router dom |
|  | // d. useRef //Nge reference element |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 13. Apa yang akan terjadi apabila kita melakukan update data state didalam componentDidUpdate? |
|  | // a. Error, karena component did update tidak memperbolehkan update data didalam state |
|  | // x Infinite loop, karena update state akan mentrigger component did update |
|  | // sehingga update state dijalankan ulang dan akan mentrigger component did update kembali |
|  | // c. Update data berhasil namun component tidak akan auto re-render |
|  | // d. Update data berhasil dan component akan auto re-render |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 14. Bagaimana cara passing data antara component parent ke child didalam ReactJS? |
|  | // a. Menggunakan sebuah state global yang dapat diakses oleh semua component |
|  | // b. Menggunakan local state di dalam sebuah component |
|  | // x Menggunakan sebuah attribute HTML custom yang bernama props |
|  | // d. Menggunakan sebuah object data |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 15. Cara menyimpan data pada state berikut didalam lifecycle method adalah? |
|  | /\* |
|  | function A(){ |
|  | const [data, setData] = useState([]) |
|  | } |
|  | \*/ |
|  | // a. componentDidMount( ){ setData(["Budi", "Anton", "Firman"]) }; |
|  | // x useEffect(( ) => setData(["Budi", "Anton", "Firman"]) ); |
|  | // c. componentDidMount( ){ this.setState({data: ["Budi", "Anton", "Firman"]}) }; |
|  | // d. useEffect(( ) => data(["Budi", "Anton", "Firman"]) ) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 16. Apakah yang perlu diperbaiki dari syntax di bawah ini? |
|  | /\* |
|  | class cards extends React.Component{ |
|  | render(){ |
|  | return <h1>{this.props.data.name}</h1> |
|  | } |
|  | } |
|  | export default cards |
|  | \*/ |
|  | // a. Membungkus element h1 dengan menggunakan () |
|  | // b. Menghilangkan kurung kurawal pada this.props.data.name |
|  | // c. Menambahkan string literal pada this.props.data.name |
|  | // x Mengubah nama component ---> Menjadi Cards |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 17. Gambarkan hasil grid berikut! |
|  | /\* |
|  | <Grid |
|  | h='200px' |
|  | templateColumns='repeat(6, 1fr)' |
|  | templateRows='repeat(3, 1fr)' |
|  | gap={4} |
|  | > |
|  | <GridItem colSpan={2} bg='tomato' /> |
|  | <GridItem rowSpan={2} colSpan={4} bg='papayawhip' /> |
|  | <GridItem colSpan={5} bg='papayawhip' /> |
|  | <GridItem colSpan={3} bg='tomato' /> |
|  | </Grid> |
|  | \*/ |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 18. Berikut ini pernyataan yang tepat mengenai JSX, kecuali? |
|  | // a. JSX singkatan dari Javascript XML |
|  | // b. Dengan JSX, kita dapat menuliskan code Javascript kedalam tag HTML |
|  | // x JSX memiliki syntax yang sama dengan HTML |
|  | // d. Semua Jawaban Benar |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 19. Pernyataan yang paling tepat mengenai state yang ada di ReactJS, kecuali? |
|  | // a. State digunakan untuk menyimpan data yang diambil dari API |
|  | // b. State dapat menyimpan data bertipe array of object |
|  | // x State dapat menyimpan data secara permanen |
|  | // d. State dapat dibuat didalam class component |
|  |  |
|  |  |
|  |  |
|  |  |
|  | // 20. Bagaimana cara kita melakukan update state dari sebuah component? |
|  | // x Menggunakan method/function untuk update state seperti this.setState atau |
|  | // function yang di-return oleh useState |
|  | // b. Kita harus memanggil useState |
|  | // c. Cukup assign value baru di variable state yang kita miliki |
|  | // d. Gunakan this.state untuk assign value baru kepada state yang kita miliki |

|  |
| --- |
| // 1. Perhatikan code berikut! |
|  | /\* |
|  | <button id='btn1' onClick={onSubmit}>Submit</button> |
|  | <button id='btn2' onClick={(event) => onSubmit(event)}>Submit</button> |
|  | \*/ |
|  | // Apakah perbedaan dari kedua button di atas? |
|  | // a. 'btn1' tidak memiliki akses ke property event saat button di klik |
|  | // x Tidak ada perbedaan |
|  | // c. 'btn1' tidak dapat memanggil function onSubmit dengan baik |
|  | // d. 'btn2' tidak dapat memanggil function onSubmit dengan baik |
|  |  |
|  |  |
|  |  |
|  | // 2. Perhatikan code berikut ini! |
|  | /\* |
|  | export default function Profile(props){ |
|  | return( |
|  | <Button {...props} /> |
|  | ) |
|  | } |
|  | \*/ |
|  | // Props apa sajakah yang akan dikirim menuju component Button? |
|  | // x Semua props |
|  | // b. Semua props yang mengalami perubahan |
|  | // c. Semua props yang tidak mengalami perubahan |
|  | // d. Semua props dari child |
|  |  |
|  |  |
|  |  |
|  | // 3. Bagaimana bisa kita mengganti apa yang ditampilkan oleh sebuah component di ReactJS? |
|  | // a. Tidak bisa, karena component di React bersifat statis |
|  | // x Membuat sebuah state yang dapat diubah, kemudian tampilkan state tersebut |
|  | // di dalam component React |
|  | // c. Membuat sebuah variable yang dapat diubah, kemudian tampilkan variable |
|  | // tersebut di dalam component React |
|  | // d. Melakukan setItem di dalam localStorage dan getItem valuenya di component React |
|  |  |
|  |  |
|  |  |
|  | // 4. Berikut ini, manakah flow yang benar ketika kita ingin menyimpan data dengan menggunakan library redux? |
|  | // x Actions > Middleware > Reducers > Store > Component |
|  | // b. Actions > Reducers > Middleware > Store > Component |
|  | // c. Actions > Middleware > Store > Component |
|  | // d. Actions > Reducers > Store > Component |
|  |  |
|  |  |
|  |  |
|  | // 5. Perhatikan code berikut! |
|  | /\* |
|  | export default function Sidebar(){ |
|  | const category = ["Drink", "Food", "Snack"] |
|  |  |
|  | const mapCategory = () => { |
|  | return category.map((item, idx) => { |
|  | return( |
|  | <div key={idx}> |
|  | {item} |
|  | </div> |
|  | ) |
|  | }) |
|  | } |
|  |  |
|  | return( |
|  | <> |
|  | {mapCategory()} |
|  | </> |
|  | ) |
|  | } |
|  | \*/ |
|  | // Apakah ada kesalahan dalam pembuatan component sidebar di atas? |
|  | // x Tidak ada |
|  | // b. Ada! Karena function mapCategory berusaha me-render 'category' yang bukan merupakan sebuah state |
|  | // c. Ada! Function mapCategory tidak akan menghasilkan apapun, karena tidak ada event listener yang memanggil |
|  | // function tersebut |
|  | // d. Ada! Karena method .map di dalam function mapCategory tidak menggunakan parameter value dan index |
|  |  |
|  |  |
|  |  |
|  | // 6. Mengapa kita membutuhkan redux ketimbang local state di dalam project React? |
|  | // x Untuk me-manage state yang lebih kompleks |
|  | // b. Untuk meningkatkan performa aplikasi ReactJS agar lebih baik |
|  | // c. Untuk menyimpan data secara permanen |
|  | // d. Untuk menyiapkan sebuah production app |
|  |  |
|  |  |
|  |  |
|  | // 7. Perhatikan code berikut! |
|  | // <Route path='/:productId' /> |
|  | // :productId disebut dengan? |
|  | // a. Rute param |
|  | // x Route parameter |
|  | // c. Route id |
|  | // d. Route link |
|  |  |
|  |  |
|  |  |
|  | // 8. Perhatikan code berikut! |
|  | /\* |
|  | export default function Popup(){ |
|  | const message = 'Register Success!' |
|  |  |
|  | return( |
|  | <div> |
|  | {message} |
|  | </div> |
|  | ) |
|  | } |
|  | \*/ |
|  | // {message} disebut dengan? |
|  | // a. JS Statement |
|  | // x JS Expression |
|  | // c. JSX Wrapper |
|  | // d. JS Logic |
|  |  |
|  |  |
|  |  |
|  | // 9. Manakah yang mendeskripsikan functional component React dengan baik? |
|  | // a. Function component selalu bersifat stateless |
|  | // x Function component menerima sebuah prop object dan akan return sebuah React element |
|  | // c. Function component dibutuhkan untuk menggantikan class component di React |
|  | // d. Function component sama saja seperti class component |
|  |  |
|  |  |
|  |  |
|  | // 10. Berikut ini manakah library yang harus kita install ketika ingin melakukan http method di dalam |
|  | // project React? |
|  | // a. fetch |
|  | // x axios |
|  | // c. redux |
|  | // d. react-router-dom |
|  |  |
|  |  |
|  |  |
|  | // 11. Pada proses development, berapakah default port project React kita pada saat dijalankan? |
|  | // x 3000 |
|  | // b. 8080 |
|  | // c. 5000 |
|  | // d. 3001 |
|  |  |
|  |  |
|  |  |
|  | // 12. Fitur manakah yang dapat menyimpan data secara permanen di sisi client/frontend? |
|  | // a. Session storage |
|  | // b. Redux |
|  | // x Local storage |
|  | // d. Local state |
|  |  |
|  |  |
|  |  |
|  | // 13. Berikut ini manakah urutan lifecycle method di dalam component React? |
|  | // a. Component Did Mount > Render > Component Did Update > Component Will Unmount |
|  | // b. Render > Component Did Update > Component Did Mount > Component Will Unmount |
|  | // x Render > Component Did Mount > Component Did Update > Component Will Unmount |
|  | // d. Semua jawaban salah |
|  |  |
|  |  |
|  |  |
|  | // 14. Bagaimana cara passing data antara component parent ke child didalam ReactJS? |
|  | // a. Menggunakan sebuah state global yang dapat diakses oleh semua component |
|  | // b. Menggunakan local state di dalam sebuah component |
|  | // x Menggunakan sebuah attribute HTML custom yang bernama props |
|  | // d. Menggunakan sebuah object data |
|  |  |
|  |  |
|  |  |
|  | // 15. Bagaimana cara menulisakan inline-style untuk font-size:12px dan color:red |
|  | // a. style={{font-size:12,color:'red'}} // font-size & 12 |
|  | // x style={{fontSize:"12px",color:'red'}} |
|  | // c. style={fontSize:"12px",color:'red'} // Kurung kurawal hanya satu |
|  | // d. style={{font-size:12px,color:'red'}} // font-size & 12 |
|  |  |
|  |  |
|  |  |
|  | // 16. Jika ingin membuat beberapa component ReactJS pada flex-box sejajar kesamping dan urutannya dibalik, attribute manakah |
|  | // yang kita digunakan ? |
|  | // a. flex-wrap = "nowrap" |
|  | // b. flexDirection = "column-reverse" |
|  | // c. flexDirection = "reverse" |
|  | // x flexDirection = "row-reverse" |
|  |  |
|  |  |
|  |  |
|  | // 17. Berikut ini element HTML manakah yang memiliki default display block? |
|  | // x h5 |
|  | // b. span |
|  | // c. input |
|  | // d. button |
|  |  |
|  |  |
|  |  |
|  | // 18. Berikut ini manakah cara yang paling tepat ketika ingin mengirimkan data yang dinamis |
|  | // menuju component sibling? |
|  | // a. Dengan membuat sebuah function di dalam file app.js, kemudian function tersebut dipanggil |
|  | // di dalam masing-masing component siblings |
|  | // b. Dengan melakukan localStoraget.setItem sebuah value dari component a, |
|  | // kemudian melakukan localStorage.getItem value di component b |
|  | // c. Dengan melakukan axios.post dari component a, kemudian melakukan axios.get di component b |
|  | // x Dengan membuat sebuah function yang men-trigger sebuah state di dalam app.js |
|  | // untuk kemudian mengirimkan props dari app.js |
|  |  |
|  |  |
|  |  |
|  | // 19. Bagaimanakah cara meng-import file css dengan benar? |
|  | // x <link rel="stylesheet" type="text/css" href="externalStylesheet.css"> |
|  | // b. <link rel="stylesheet" type="text/css" src="externalStylesheet.css"> |
|  | // c. <stylesheet src="externalStylesheet.css"></stylesheet> |
|  | // d. <stylesheet>externalStylesheet.css</stylesheet> |