Nama : Ricky alfian saputra

Nim : 312210279

Matkul : Rekayasa perangkat lunak

Kisi – Kisi UTS

1. Berikan penjelasan mengenai Rekayasa Perangkat Lunak dari mulai sejarahnya, metodenya,

dan perkembangannya!

Jawab:

Rekayasa Perangkat Lunak (Software Engineering) adalah disiplin ilmu yang berkaitan

dengan pengembangan perangkat lunak atau perangkat komputer dengan pendekatan yang

sistematis, terstruktur, dan disiplin. Ini mencakup seluruh siklus hidup perangkat lunak,

mulai dari perencanaan hingga pemeliharaan. Di bawah ini, saya akan memberikan

penjelasan singkat tentang sejarah, metode, dan perkembangan rekayasa perangkat lunak:

Sejarah:

• Era Awal (Sebelum 1960): Pada awalnya, pengembangan perangkat lunak cenderung

tidak terstruktur dan berbasis pada metode ad hoc. Pada periode ini, pengembangan

perangkat lunak umumnya dilakukan oleh ilmuwan komputer yang juga merancang

perangkat keras.

• Krisis Perangkat Lunak (1960-an): Dalam tahun 1960-an, terjadi "krisis perangkat

lunak" karena semakin kompleksnya perangkat lunak yang dikembangkan. Hal ini

memicu pencarian metode yang lebih sistematis.

• Konsep Rekayasa Perangkat Lunak (1960-an - 1970-an): Pada periode ini, muncul

konsep rekayasa perangkat lunak, dan metode seperti Waterfall Model dan Structured

Programming muncul untuk membantu mengelola proyek perangkat lunak.

• Objek-Oriented Programming (OOP) (1980-an): Metode berorientasi objek, seperti

bahasa pemrograman C++ dan Java, menjadi populer dan mempengaruhi pendekatan

rekayasa perangkat lunak.

• Gerakan Agile (2000-an): Gerakan Agile berkembang dan menggandeng metodologi

seperti Scrum, Kanban, dan Extreme Programming (XP) yang fokus pada fleksibilitas

dan kolaborasi tim.

Metode:

• Waterfall Model: Model ini menggambarkan siklus hidup perangkat lunak sebagai

aliran linear dari tahap analisis, perancangan, implementasi, pengujian, hingga

pemeliharaan. Setiap tahap harus selesai sebelum memasuki tahap berikutnya.

• Iterative and Incremental Development: Pendekatan ini membagi pengembangan

perangkat lunak menjadi iterasi pendek yang masing-masing menambahkan

fungsionalitas tambahan. Iterasi ini seringkali mengikuti siklus yang berulang.

• Agile Development: Pendekatan ini menekankan kolaborasi tim yang intensif,

perubahan iteratif, dan respons cepat terhadap perubahan kebutuhan. Metodologi Agile

seperti Scrum, Kanban, dan XP sangat populer dalam pengembangan perangkat lunak.

• Model Spiral: Model Spiral menggabungkan elemen iteratif dari pengembangan dengan

elemen evaluasi risiko. Setiap iterasi melibatkan perencanaan, analisis risiko,

pengembangan, dan evaluasi.

Perkembangan Terkini:

• Penggunaan Kecerdasan Buatan (AI) dan Machine Learning: Teknologi AI dan

Machine Learning semakin banyak digunakan dalam pengembangan perangkat lunak

untuk membuat sistem yang lebih pintar dan adaptif.

• DevOps: DevOps adalah pendekatan yang menggabungkan pengembangan perangkat

lunak (Dev) dengan operasi (Ops) untuk meningkatkan efisiensi, responsifitas, dan

kualitas pengembangan serta operasi perangkat lunak.

• Containerization dan Mikro Layanan: Penggunaan kontainer seperti Docker dan

pendekatan mikro layanan semakin populer untuk memudahkan penyebaran dan

skalabilitas aplikasi.

• Pendekatan No-Code/Low-Code: Alat no-code/low-code memungkinkan orang yang

tidak memiliki latar belakang teknis mendesain dan mengembangkan perangkat lunak

dengan mudah.

Rekayasa perangkat lunak terus berkembang seiring dengan perubahan dalam

teknologi, metode, dan tuntutan pasar. Mendorong efisiensi dan kualitas pengembangan

perangkat lunak adalah tujuan utama, dan pendekatan yang sesuai harus dipilih berdasarkan

karakteristik proyek dan tujuan bisnis.

2. Jekaskan mengenai Analisa dan Perancangan Sistem dengan pendekatan terstruktur!

Jawab:

Analisis dan Perancangan Sistem (APSI) dengan pendekatan terstruktur adalah

metodologi yang digunakan dalam pengembangan perangkat lunak dan sistem informasi

untuk merancang sistem dengan struktur yang terorganisir, hierarkis, dan terperinci.

Pendekatan ini fokus pada pemisahan tugas dalam suatu sistem menjadi modul atau

komponen yang lebih kecil.

3. Jekaskan mengenai Analisa dan Perancangan Sistem dengan pendekatan berorientasi objek!

Jawab:

Analisis dan Perancangan Sistem (APSI) dengan pendekatan berorientasi objek

adalah metode yang digunakan dalam pengembangan perangkat lunak dan sistem informasi

untuk merancang sistem dengan mempertimbangkan objek-objek sebagai entitas dasar yang

meliputi objek dalam kehidupan nyata, enkapsulasi untuk menjaga integrasi data, pewarisan,

polimorfism dan diagram UML.

4. Berikan penjelasan mengenai UML!

Jawab:

Unified Modeling Language (UML) adalah sebuah bahasa visual yang digunakan

untuk memodelkan, merancang, dan mendokumentasikan perangkat lunak dan sistem

informasi. UML adalah standar industri yang diterima secara luas dan digunakan oleh

pengembang perangkat lunak, analis sistem, dan arsitek perangkat lunak untuk

menggambarkan struktur, interaksi, dan perilaku sistem. Ini membantu dalam komunikasi

yang lebih baik antara para pemangku kepentingan proyek dan memfasilitasi pemahaman

yang lebih baik tentang sistem yang akan dibangun.

5. Berikan penjelasan mengeani program disain!

Jawab:

Program desain adalah perangkat lunak yang digunakan untuk menciptakan,

mengedit, dan merancang berbagai jenis proyek visual, termasuk grafis, ilustrasi, tata letak,

dan desain produk. Program desain ini sangat berguna dalam berbagai industri seperti desain

grafis, arsitektur, desain interior, desain mode, dan banyak lagi.

6.Tuliskan nama proyek anda

Jawab:

Proyek: Aplikasi Pengingat Kesehatan

7. Tuliskan SDLC planning pada tahap perancangan atau desain:

a. Pemersalahan: Individu seringkali lupa untuk menjaga kebiasaan hidup sehat seperti minum air secara teratur, berolahraga, dan mengonsumsi obat-obatan yang penting.

Solusi: Mengembangkan aplikasi mobile yang memberikan pengingat dan saran seputar kesehatan, termasuk minum air, jadwal olahraga, dan pengingat konsumsi obat-obatan.

Metode Pengembangan:

Metode Agile Development cocok untuk proyek seperti ini, karena memungkinkan fleksibilitas dalam menanggapi perubahan kebutuhan pengguna selama pengembangan berlangsung.

Tahapan Pengembangan (Metode Agile):

Pembentukan Tim dan Penetapan Tujuan: Tim pengembang, desainer, dan ahli kesehatan membentuk tim lintas-fungsional dan mendefinisikan tujuan proyek.

Penyusunan Daftar Fitur dan Prioritasi: Mengidentifikasi fitur utama seperti pengingat minum air, pengingat olahraga, dan pengingat obat. Prioritaskan berdasarkan urgensi dan kepentingan.

Siklus Iteratif (Sprint): Setiap sprint berlangsung selama 2-4 minggu, tim memilih sejumlah fitur untuk diimplementasikan dan membangunnya.

Pengujian dan Evaluasi Pengguna (Testing and User Evaluation): Setelah setiap sprint, fitur yang baru diimplementasikan diuji dan dievaluasi oleh tim pengujian dan pengguna.

Retrospesktif Sprint (Sprint Retrospective): Setelah setiap sprint, tim melakukan evaluasi retrospektif untuk memperbaiki proses pengembangan.

Dengan pendekatan Agile Development, tim dapat secara terus-menerus menyesuaikan dan

memperbaiki aplikasi berdasarkan umpan balik pengguna dan memastikan bahwa aplikasi memenuhi kebutuhan kesehatan mereka.

b.Software dan Hardware:

Software:

-IDE seperti Android Studio (untuk Android) atau Xcode (untuk iOS).

-Bahasa pemrograman seperti Java/Kotlin (untuk Android) atau Swift (untuk iOS).

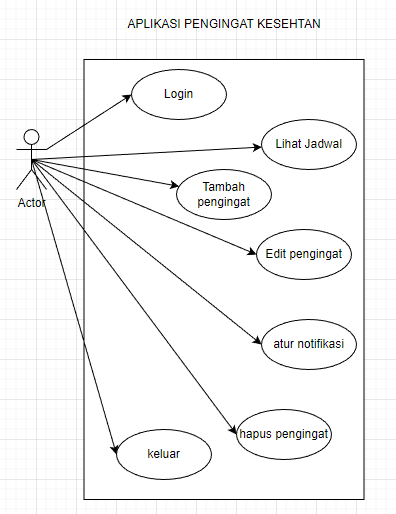
-Alat desain seperti Adobe XD atau Sketch.

Hardware:

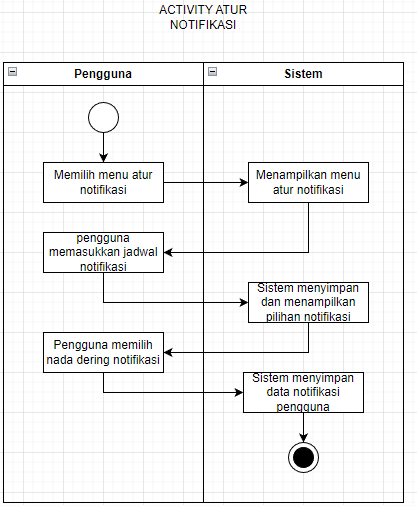
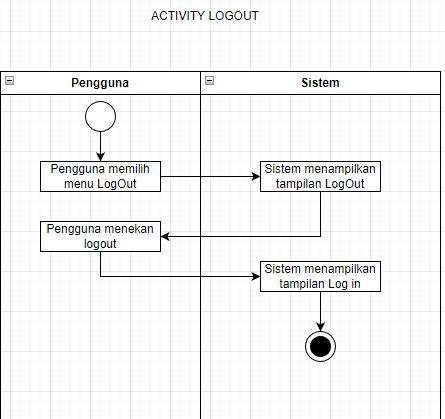
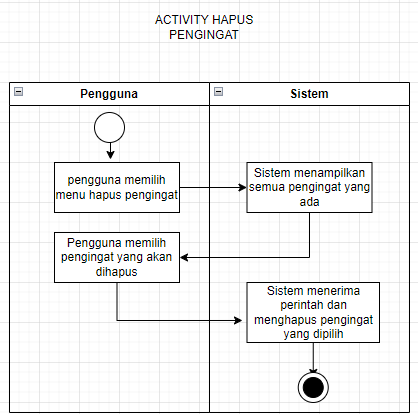
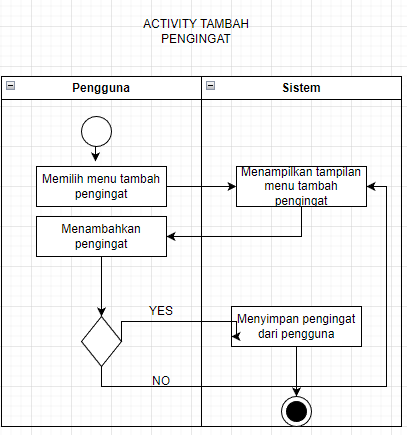
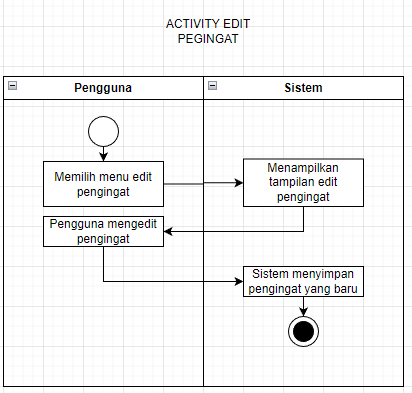
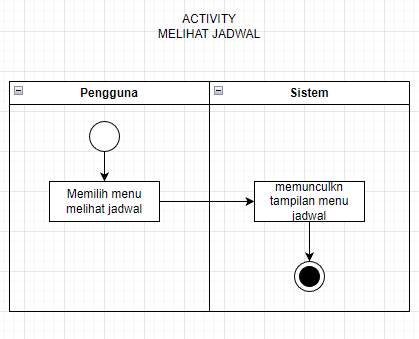
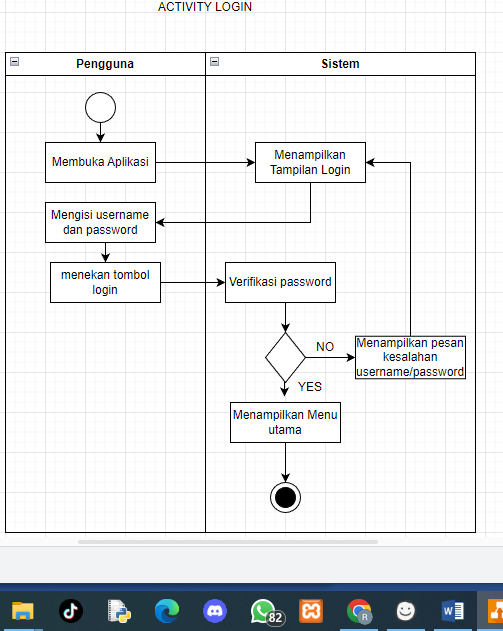
-Komputer atau laptop untuk pengembangan aplikasi. Perangkat Android atau iOS untuk pengujian dan debugging.

-Untuk pengujian lanjutan, mungkin diperlukan berbagai jenis perangkat seluler.

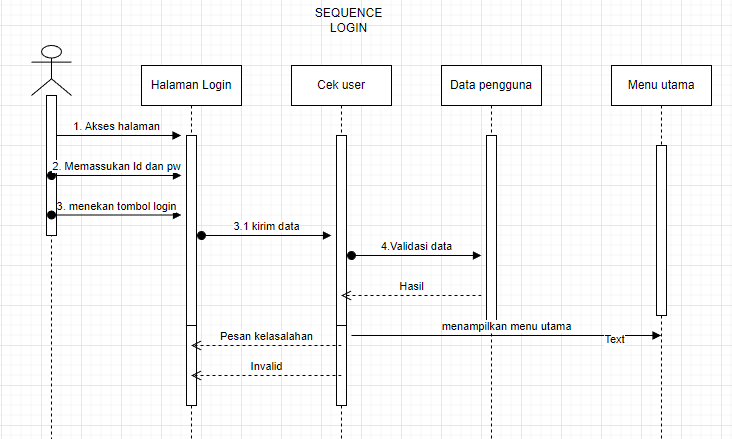
c. Usecase diagram

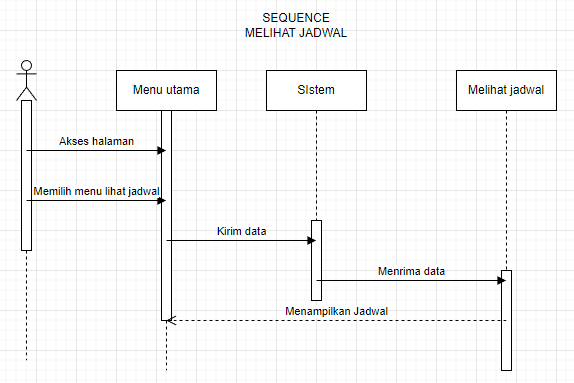


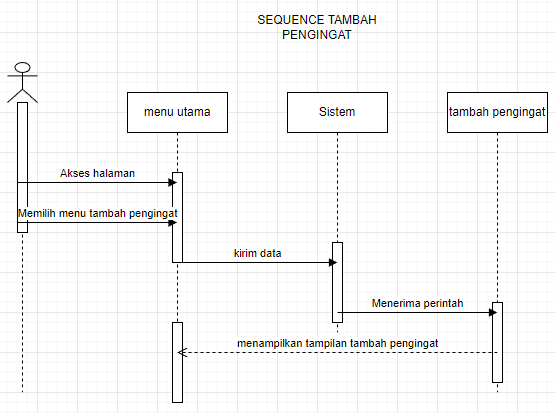
d.actvity diagram

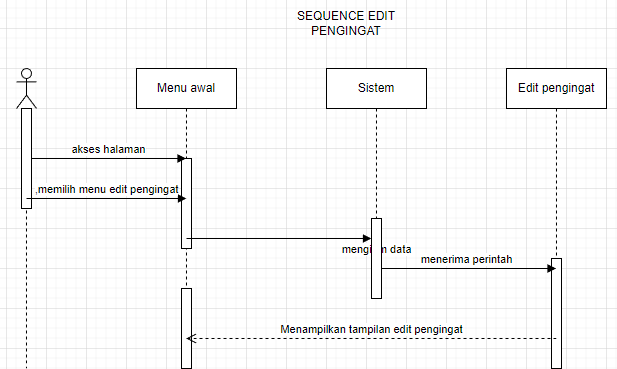


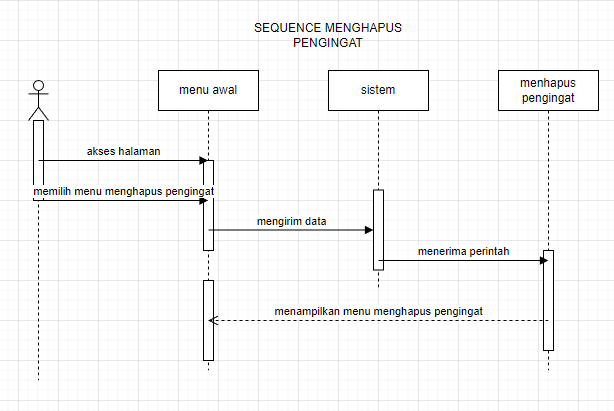
d.sequence diagram

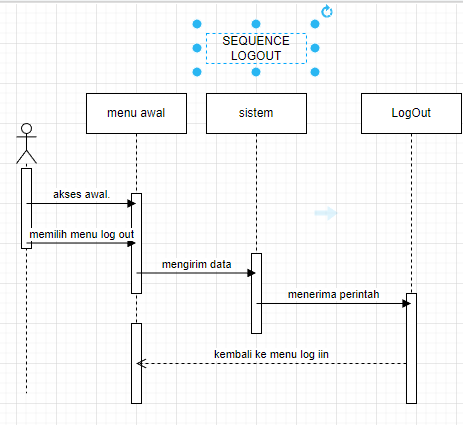
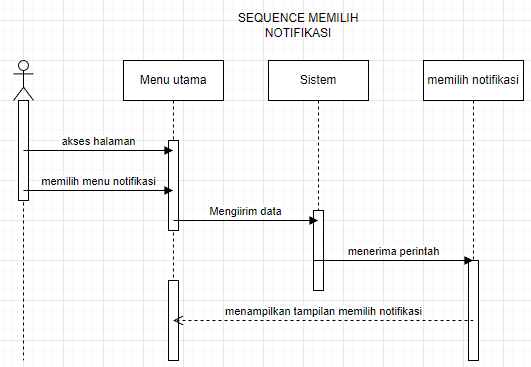




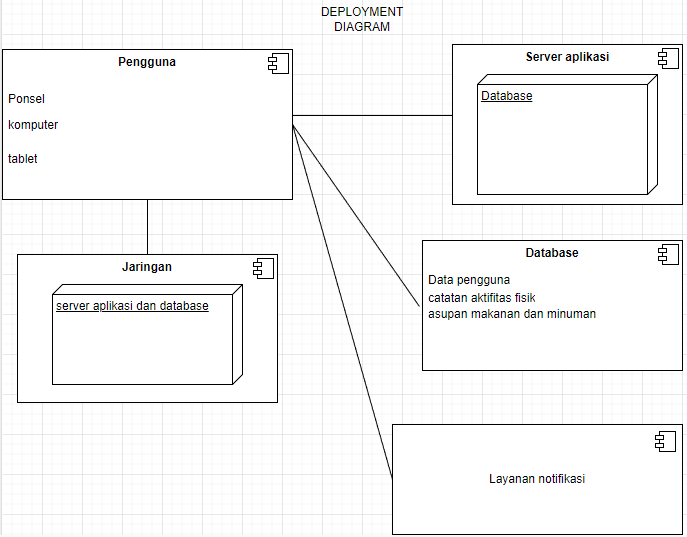
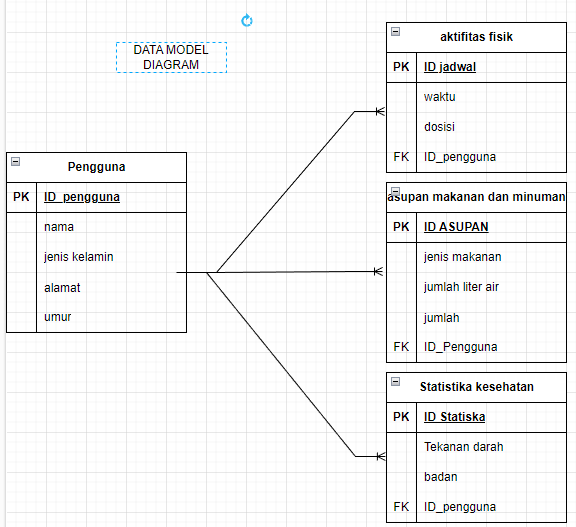
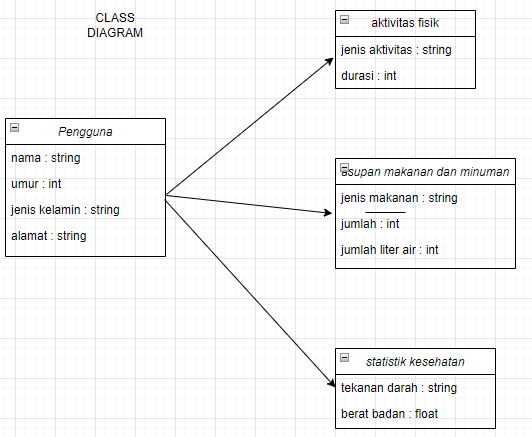








8.Tahap perancangan atau desain



USER INTERFACE

