

LAPORAN AKHIR
MAGANG INDUSTRI / MAGANG dan STUDI
INDEPENDEN BERSERTIFIKAT
RANCANG BANGUN APLIKASI Di PT REKAINDO
GLOBAL JASA

Diajukan untuk memenuhi persyaratan kelulusan
Program MSIB/Magang MBKM

oleh :
Muhammad Rayhan Gibran /2141762099



PROGRAM STUDI D4 SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2024

LEMBAR PENGESAHAN

Yang bertandatangan dibawah ini

Nama : Hisyam Syafiq Almughni Ahmad
NIP : 942200003
Jabatan : Manajer QMSHE & TI

Menerangkan bahwa mahasiswa

Muhammad Rayhan Gibran NIM. 2141762099

Telah menyelesaikan Magang/MBKM di

Nama Perusahaan : PT. Rekindo Global Jasa
Alamat Perusahaan : Jl. Candi Sewu No.30, Manguharjo, Kota Madiun
Bidang : QM & SHE (IT dan K3)
Waktu Pelaksanaan : 15 Juli 2024 – 15 Desember 2024

Madiun,

Hisyam Syafiq Almughni Ahmad

NIP. 942200003

LEMBAR PENGESAHAN

Laporan MSIB/Magang MBKM

**RANCANG BANGUN APLIKASI
DI PT REKAINDO GLOBAL JASA**

telah disetujui dan disahkan pada presentasi Laporan MSIB/Magang MBKM

Jurusan Teknologi Informasi

Politeknik Negeri Malang

Pada tanggal

Malang,

Mengetahui dan Menyetujui,

Ketua Program Studi D-IV

Dosen Pembimbing

Hendra Pradibta, S.E., M.Sc.
NIP. 198305212006041003

Usman Nurhasan, S.Kom., M.T.
NIP. 198609232015041001

Mengetahui,

Ketua Jurusan Teknologi Informasi,

Dr. Eng. Rosa Andrie Asmara, S.T., M.T.
NIP. 198010102005011001

Abstraksi

Rekachain merupakan sistem berbasis website dan mobile yang mengakomodasi produksi panel kereta api mulai dari data pengerjaan panel, data trainset, data material, dan data pengerjaan. Sistem ini berbasis website yang dapat berjalan pada desktop, maupun telepon seluler, yang memiliki web browser dan terhubung secara online ke internet. Rekachain dirancang untuk mengatasi berbagai tantangan produksi barang yang dihadapi PT. Rekindo Global Jasa, khususnya dalam ranah produksi. Tantangan utama yang dihadapi adalah koordinasi dan transparansi antara berbagai pihak yang tergabung dalam proses produksi komponen pendukung kereta api. Dengan Rekachain, perusahaan dapat memantau kemajuan tahap produksi di setiap lini produksi, sehingga memberikan transparansi.

Kata kunci : PT. Rekindo Global Jasa, Rekachain, Produksi panel kereta api

Kata Pengantar

Puji syukur kehadiran Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga saya dapat menyelesaikan program magang industri ini dengan baik. Program ini telah memberikan saya kesempatan untuk belajar, berkembang, dan mendapatkan pengalaman dalam dunia kerja. Penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada berbagai pihak yang telah mendukung dan memfasilitasi pelaksanaan program ini:

1. Allah SWT yang telah memberikan kesehatan, kelancaran, kemudahan dalam melaksanakan magang industry ini.
2. Nabi Muhammad SAW yang telah mengajarkan tentang bagaimana menjalani kehidupan.
3. Keluarga, yang senantiasa memberikan dukungan doa dan material sehingga penulis dapat menyelesaikan magang industri dengan baik.
4. Bapak Usman Nurhasan, S.Kom., M.T. selaku pembimbing akademik.
5. Teman – teman terdekat yang memberikan bantuan, saran untuk melaksanakan program ini.

Penulis berharap pengalaman dan ilmu yang didapatkan selama mengikuti program ini dapat menjadi bekal yang bermanfaat bagi karier penulis di masa depan. Penulis juga berharap program ini dapat terus berlanjut dan memberikan dampak positif kepada lebih banyak mahasiswa. Semoga segala bantuan, dukungan, dan kebaikan yang telah diberikan oleh semua pihak mendapatkan balasan yang setimpal dari Tuhan Yang Maha Esa.

Daftar Isi

LEMBAR PENGESAHAN.....	1
LEMBAR PENGESAHAN.....	2
BAB I PENDAHULUAN.....	9
I.1 Latar belakang.....	9
I.2 Lingkup.....	10
I.3 Tujuan.....	10
BAB II LINGKUNGAN ORGANISASI PT REKAINDO GLOBAL JASA..	11
2.1. Struktur Organisasi.....	11
Gambar 2.1 Struktur organisasi.....	11
2.2. Lingkup Pekerjaan.....	11
2.3. Deskripsi Pekerjaan.....	11
2.4. Jadwal Kerja.....	12
BAB III PEMBUATAN APLIKASI REKACHAIN.....	13
3.1. Rekachain.....	13
3.2. Proses Pembuatan Aplikasi Rekachain.....	13
Gambar 3.1 Gambaran kasar proses produksi barang.....	14
Gambar 3.2 Cuplikan Use Case Diagram.....	15
Gambar 3.3 Cuplikani kode Kotlin.....	15
3.3 Pencapaian Hasil.....	15
BAB IV HASIL PROYEK MAGANG.....	17
4.1 Sistem yang dibuat.....	17
4.2 Implementasi sistem di industri.....	39
4.3 Hasil yang didapat selama magang.....	40
BAB V PENUTUP.....	41
5.1. Kesimpulan.....	41
5.2. Saran.....	41
LAMPIRAN A. TERM OF REFERENCE (TOR).....	43
1. Latar Belakang.....	43
LAMPIRAN B. LOG ACTIVITY.....	45
LAMPIRAN C. DOKUMEN TEKNIK.....	46
1. Proses Kelola Proyek.....	49
2. Proses Kelola Gerbong.....	52
3. Proses Kelola Trainset.....	55
4. Proses Kelola Panel.....	56
5. Proses Kelola Komponen Panel.....	61
6. Proses Cetak QR.....	64
7. Proses Dashboard Proyek.....	66
8. Proses Supervisor Mekanik dan Elektrik.....	66

9. Proses Mekanik / Elektrik.....	69
10. Proses Assembly.....	71
11. Proses QC Proses.....	73
12. Proses QC Final.....	75

DAFTAR GAMBAR

Gambar 3.1 Gambaran kasar proses produksi barang.....	14
Gambar 3.2 Cuplikan Use Case Diagram.....	15
Gambar 3.3 Cuplikani kode Kotlin.....	15
Gambar 4.1 Halaman Login.....	17
Gambar 4.2 Halaman lupa password.....	18
Gambar 4.3 Halaman Dashboard.....	18
Gambar 4.4 Halaman Pemindaian.....	19
Gambar 4.5 Halaman data KPM trainset.....	19
Gambar 4.6 Halaman data KPM panel.....	20
Gambar 4.8 Halaman konfirmasi pekerja.....	21
Gambar 4.9 Halaman Profil.....	21
Gambar 4.10 Halaman data akun.....	22
Gambar 4.12 Halaman riwayat pekerjaan level PPC.....	23
Gambar 4.13 Halaman riwayat pekerjaan level Supervisor.....	23
Gambar 4.14 Halaman riwayat pekerjaan level worker.....	24
Gambar 4.15 Halaman riwayat pekerjaan level Worker QC.....	24
Gambar 4.16 Halaman kerja level Worker PPC.....	25
Gambar 4.17 Halaman kerja level supervisor.....	26
Gambar 4.20 Halaman pekerjaan ditolak level worker.....	27
Gambar 4.21 Halaman catatan tidak lolos level QC.....	28
Gambar 4.22 Cuplikan kode halaman login.....	29
Gambar 4.23 Cuplikan kode halaman lupa password.....	30
Gambar 4.24 Cuplikan kode halaman dashboard.....	30
Gambar 4.25 Cuplikan kode halaman data kartu penarikan material.....	31
Gambar 4.27 Cuplikan kode halaman konfirmasi pekerja.....	33
Gambar 4.28 Cuplikan kode halaman profil.....	34
Gambar 4.29 Cuplikan kode halaman akun.....	34
Gambar 4.30 Cuplikan kode halaman ganti password.....	35
Gambar 4.31 Cuplikan kode halaman riwayat pekerjaan.....	35
Gambar 4.32.1 Cuplikan kode halaman kerja Supervisor.....	36
Gambar 4.33.2 Cuplikan kode halaman kerja Worker.....	37
Gambar 4.34.3 Cuplikan kode halaman Quality Control.....	38

Gambar 4.35 Dokumentasi.....	39
Gambar C.1 Use Case Diagram.....	48
Gambar C.2 Class Diagram.....	48
Gambar C.3 Entity Relationship Diagram.....	48

BAB I PENDAHULUAN

I.1 Latar belakang

Manajemen informasi produksi yang efisien adalah salah satu kunci keberhasilan operasional sebuah perusahaan manufaktur. PT. Rekaindo Global Jasa, sebagai perusahaan manufaktur yang menyuplai komponen pendukung kereta api untuk PT. INKA, menyadari pentingnya mengadopsi sistem informasi produksi untuk meningkatkan efisiensi. Sistem informasi tersebut akan dinamakan Rekachain.

Rekachain dirancang untuk mengatasi berbagai tantangan dalam aktivitas yang dihadapi PT. Rekaindo Global Jasa, khususnya dalam ranah produksi. Tantangan utama yang dihadapi adalah koordinasi dan transparansi antara berbagai pihak yang tergabung dalam proses produksi komponen pendukung kereta api. Dengan Rekachain, perusahaan dapat memantau kemajuan tahap produksi di setiap lini produksi, sehingga memberikan transparansi dan mengurangi risiko keterlambatan.

Lebih jauh lagi, Rekachain juga mendukung upaya PT. Rekaindo Global Jasa dalam memenuhi standar kualitas tinggi yang pada hasil komponen perkeretaapian. Dengan kemampuan untuk melacak setiap komponen dari sumber material hingga hasil produk akhir, perusahaan dapat memastikan bahwa setiap komponen memenuhi spesifikasi yang diinginkan dan mengidentifikasi masalah kualitas dengan cepat jika terjadi.

Dalam rangka menghadapi tantangan di masa depan dan memastikan keberlanjutan bisnis, implementasi Rekachain sebagai sistem informasi supply chain management menjadi sangat mendesak. Dengan Rekachain, PT. Rekaindo Global Jasa tidak hanya dapat memperkuat posisinya sebagai pemasok terpercaya bagi PT. INKA.

I.2 Lingkup

Lingkup magang di PT Rekindo Global Jasa yaitu penulis terlibat dalam pengembangan aplikasi berbasis *mobile*, yang mencakup perencanaan, desain, pembangunan/pengembangan, dan pengujian.

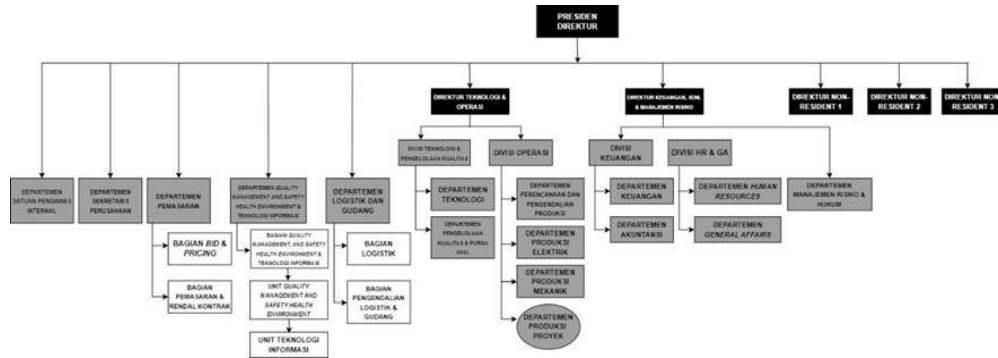
I.3 Tujuan

Tujuan menjalani magang di PT Rekindo Global Jasa adalah :

1. Mendapatkan pengalaman praktis dalam lingkungan dunia kerja dengan menerapkan pengetahuan teoritis yang dipelajari baik di dalam maupun luar perguruan tinggi.
2. Mengembangkan ketrampilan dalam pengembangan *website* dan *mobile* serta *soft skills* seperti komunikasi dan kerjasama tim.

BAB II LINGKUNGAN ORGANISASI PT REKAINDO GLOBAL JASA

2.1. Struktur Organisasi



Gambar 2.1 Struktur organisasi

Struktur organisasi PT. Rekaindo Global Jasa terdiri dari kurang lebih 4 Divisi seperti Divisi Teknologi & Pengelolaan Kualitas, Divisi Operasi, Divisi Keuangan, Divisi HR & GA dan kurang lebih 16 departemen seperti Departemen Satuan Pengawas Internal, Departemen Sekretaris Perusahaan, Departemen Pemasaran, Departemen Quality Management And Safety Health Environment & Teknologi Informasi, Departemen Logistik dan Gudang dll. Disini penulis di tempatkan pada departemen Quality Management And Safety Health Environment & Teknologi Informasi.

2.2. Lingkup Pekerjaan

Bagian pada PT. Rekaindo Global Jasa tempat penulis melaksanakan magang industri yaitu departement Quality Management And Safety Health Environment & Teknologi Informasi atau QM & SHE (IT dan K3), pada bagian ini lingkup pekerjaan penulis befokus pada pengembangan aplikasi berbasis android, dengan menggunakan teknologi bahasa pemrograman Kotlin.

2.3. Deskripsi Pekerjaan

2.4. Jadwal Kerja

1. Bulan Juli 2024 : Analisis Persyaratan
 - Wawancara
 - Survei Lokasi
 - Pengumpulan data
 - Identifikasi lingkup pekerjaan
2. Bulan Agustus 2024 : Penyusunan spesifikasi kebutuhan dan Dokumen SKPL
 - Pembuatan alur proses bisnis
 - Penyusunan dokumen SKPL
 - Pembuatan Unified Modeling
3. Bulan September - Oktober 2024 : Implementasi
 - Implementasi Back-End
 - Implementasi Front-End
 - Implementasi Front-End Mobile
4. Bulan November - Desember 2024 : Pengujian & Deployment
 - Pengujian aplikasi web
 - Pengujian mobile
 - Deployment

BAB III PEMBUATAN APLIKASI REKACHAIN

3.1. Rekachain

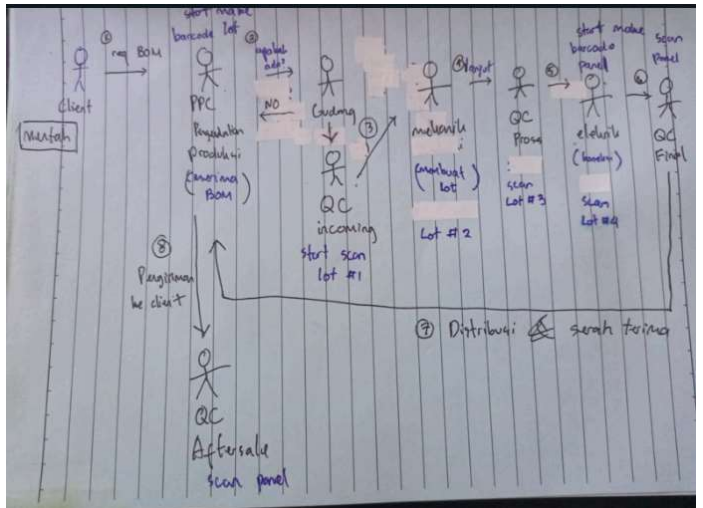
Rekachain merupakan sistem berbasis website dan mobile yang mengakomodasi produksi panel kereta api mulai dari data pengerjaan panel, data trainset, data material, dan data pengerjaan. Sistem ini berbasis website yang dapat berjalan pada desktop, maupun telepon seluler, yang memiliki web browser dan terhubung secara online ke internet. Rekachain dirancang untuk mengatasi berbagai tantangan dalam rantai pasok yang dihadapi PT. Rekindo Global Jasa, khususnya dalam ranah produksi. Tantangan utama yang dihadapi adalah koordinasi dan transparansi antara berbagai pihak yang tergabung dalam proses produksi komponen pendukung kereta api. Dengan Rekachain, perusahaan dapat memantau kemajuan tahap produksi di setiap lini produksi, sehingga memberikan transparansi dan mengurangi risiko keterlambatan.

3.2. Proses Pembuatan Aplikasi Rekachain

Proses yang dilakukan untuk pembuatan aplikasi Rekachain adalah sebagai berikut :

1. Pemahaman Proses Produksi

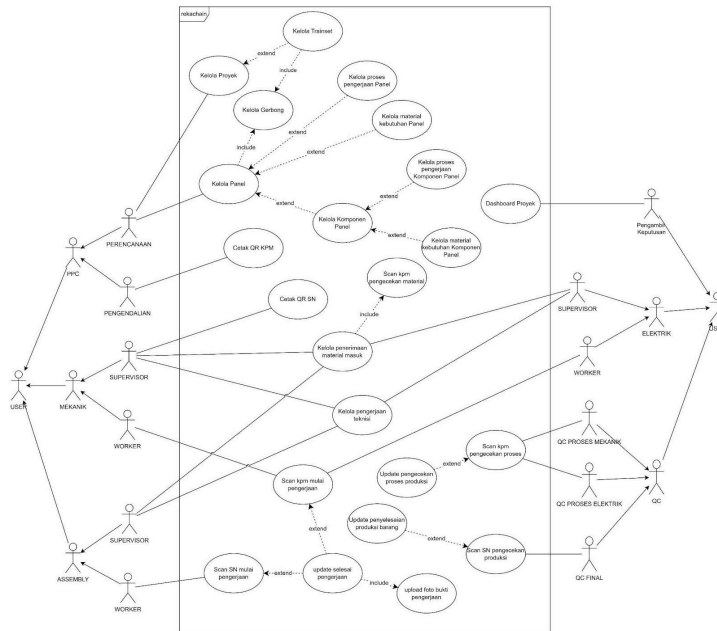
Pemahaman proses bisnis merupakan tahap awal aktivitas magang. proses ini bertujuan untuk memahami proses produksi yang diterapkan pada PT. Rekindo Global jasa.



Gambar 3.1 Gambaran kasar proses produksi barang

2. Perancangan Dokumen Spesifikasi Kebutuhan Perangkat Lunak

Perancangan spesifikasi perangkat lunak merupakan tahap penentuan fitur apa saja yang ada dalam aplikasi. proses ini bertujuan untuk mengetahui kebutuhan fungsional dari sistem yang akan dikembangkan. Dokumen terdiri dari desain database, use case diagram, sequence diagram, dan lain lain.



Gambar 3.2 Cuplikan Use Case Diagram

3. Pengembangan aplikasi

Pada tahap pengembangan aplikasi, tim Rekachain dibagi menjadi beberapa bagian yaitu Front End Web Developer, Front End Mobile Developer, Backend Developer. Penulis fokus pada bagian Front End Mobile Developer menggunakan bahasa pemrograman Kotlin.

```
private lateinit var binding: FragmentHistoryBinding
private lateinit var chipGroup: ChipGroup
private lateinit var listHistorySpvAdapter: ListHistorySpvAdapter
private lateinit var listHistoryAssyWorkerAdapter: ListHistoryAssyWorkerAdapter
private lateinit var listHistoryWorkerAdapter: ListHistoryWorkerAdapter
private lateinit var listHistoryPpcPanelAdapter: ListHistoryPpcPanelAdapter
private lateinit var listHistoryPpcTrainsetAdapter: ListHistoryPpcTrainsetAdapter
private lateinit var historyViewModel: HistoryViewModel
private lateinit var tokenHandler: TokenHandler
private lateinit var loginResponse: LoginResponse
private var currentFilter:String = AttachmentStatusEnum.ALL.status
private var job: Job? = null
var progressDialog : Dialog? = null

override fun onCreateView( inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View {
    binding = FragmentHistoryBinding.inflate(inflater, container, attachToParent false)
    return binding.root
}

override fun onViewCreated( view: View, savedInstanceState: Bundle?) {
    super.onViewCreated( view, savedInstanceState)

    chipGroup = binding.chipGroup
    tokenHandler = TokenHandler( requireContext() )

    val user = tokenHandler.getLoginResponse()
    loginResponse = Gson().fromJson( user, LoginResponse::class.java )

    val attachmentWorkerRepository = AttachmentWorkerRepository( RetrofitClient.getInstance( requireContext(), tokenHandler ) )
    historyViewModel = HistoryViewModel( attachmentWorkerRepository, tokenHandler, requireContext() )

    initRecyclerView() // set adapter()
    initView() // set view loading
    setupChipGroup() // set Chip
    observeViewModel()
}
```

Gambar 3.3 Cuplikani kode Kotlin

4. Pengujian

Pengujian merupakan proses untuk memverifikasi dan menguji kualitas serta fungsi dari sebuah aplikasi. Pengujian aplikasi merupakan bagian penting dalam siklus pengembangan perangkat lunak.

3.3 Pencapaian Hasil

Hasil dari pelaksanaan magang ini adalah Sistem Rekachain yang terdiri dari aplikasi berbasis web menggunakan framework Laravel dan aplikasi berbasis mobile menggunakan bahasa pemrograman Kotlin. Adapun dalam aplikasi ini memiliki beberapa peran yaitu :

1. Superadmin

Peran Superadmin merupakan peran paling tinggi yang memiliki akses ke hampir semua fitur yang ada di Sistem Rekachain seperti akses dalam fitur Manajemen Staf, fitur Manajemen Hak Akses, fitur manajemen daftar langkah manufaktur dll.

2. PPC

PPC terdiri dari dua jenis yaitu PPC Perencanaan yang bertugas dalam Distribusi material serta perencanaan dan pengawasan proses produksi dan PPC Pengendalian yang bertugas dalam Penyerahan material kepada pekerja (mekanik, elektrik, assembly), serta membuat request kpm baru apabila terdapat kekurangan material.

3. Supervisor

Supervisor yang bertugas dalam Pengawasan kerja karyawan yang memiliki akses dalam approve pekerjaan, approve penarikan material. Supervisor terdiri dari 3 jenis yaitu Supervisor Mekanik, Supervisor Elektrik, Supervisor Assembly .

4. Worker

Worker yang bertugas dalam produksi barang. Worker terdiri dari 3 jenis yaitu Worker Mekanik, Worker Elektrik, Worker Assembly. Worker memiliki tugas dalam hal produksi barang.

5. Quality Control (QC)

Quality Control bertugas dalam memeriksa, menguji, dan memantau kualitas produk atau layanan agar memenuhi standar yang ditetapkan. Quality Control dibagi menjadi 3 jenis QC Mekanik, QC Elektrik, QC Assembly .

BAB IV HASIL PROYEK MAGANG

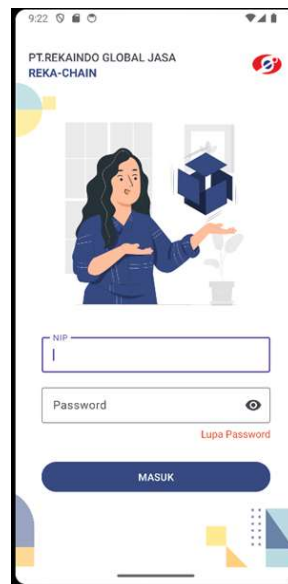
4.1 Sistem yang dibuat

Aplikasi yang dikembangkan penulis merupakan aplikasi mobile berbasis android menggunakan Bahasa Kotlin, aplikasi mobile ini memiliki peran utama untuk mengirimkan atau mengubah status produksi suatu produk. Berikut halaman yang ada dalam aplikasi beserta potongan kode aplikasi mobile Rekachain :

a. Tampilan antar muka

1. Halaman Login

Halaman login adalah halaman autentikasi pengguna, untuk memastikan yang memakai aplikasi adalah orang yang memiliki akses.

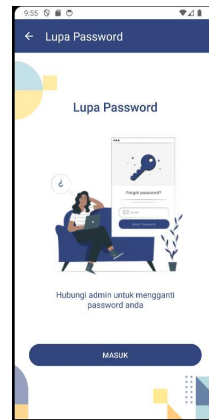


Gambar 4.1 Halaman Login

Ketika user melakukan login maka aplikasi akan menerima token, dan token tersebut digunakan untuk mengakses data data yang lain.

2. Halaman Lupa Password

Halaman lupa password adalah halaman yang digunakan untuk memberikan informasi bahwa jika lupa password maka dapat menghubungi admin.

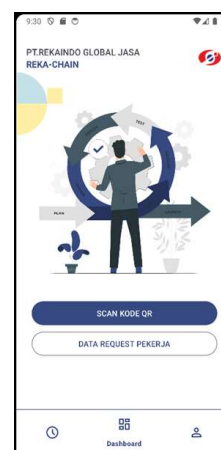


Gambar 4.2 Halaman lupa password

3. Halaman Dashboard

3.1. Dashboard Supervisor

Halaman dashboard supervisor adalah halaman untuk supervisor.

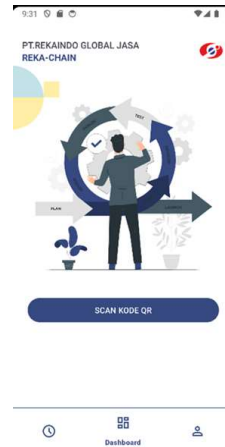


Gambar 4.3 Halaman Dashboard

Halaman dashboard adalah halaman awal ketika pengguna melakukan login.

3.2. Dashboard Selain Supervisor

Halaman selain supervisor adalah halaman yang digunakan untuk role selain supervisor.

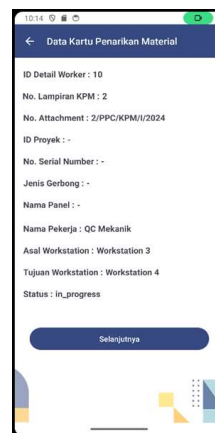


Gambar 4.4 Halaman Pemindaian

4. Halaman Data Kartu Penarikan Material

5.1. KPM Train Set

Halaman KPM trainset adalah halaman yang menampilkan data KPM trainset.



Gambar 4.5 Halaman data KPM trainset

Halaman ini mendapat data setelah pengguna melakukan scan.

5.2. KPM Panel

Halaman KPM panel adalah halaman yang menampilkan data KPM Panel.



Gambar 4.6 Halaman data KPM panel

5. Halaman Data Request Pekerja

Halaman Data Request Pekerja adalah halaman yang menampilkan data request pekerja yang ditampilkan pada role supervisor.



Gambar 4.7 Halaman data request pekerja

Halaman data request mempunyai fitur untuk melakukan filter berdasarkan status.

6. Halaman Konfirmasi Pekerja

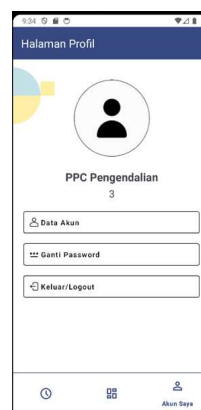
Halaman Konfirmasi Pekerja adalah halaman yang digunakan supervisor untuk menerima atau menolak request kerja oleh worker.



Gambar 4.8 Halaman konfirmasi pekerja

7. Halaman Profil

Halaman Profil adalah halaman yang digunakan oleh user untuk melihat data akun, mengganti password maupun Logout

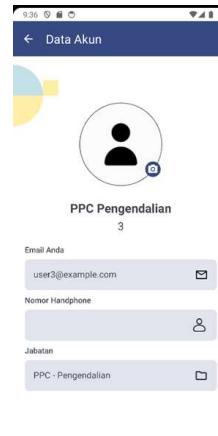


Gambar 4.9 Halaman Profil

Halaman profil ini terdapat gambar profil pengguna yang dapat diganti.

8. Halaman Data Akun

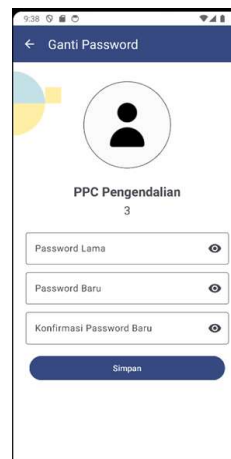
Halaman data akun adalah halaman yang digunakan untuk melihat data detail akun pengguna.



Gambar 4.10 Halaman data akun

9. Halaman Ganti Password

Halaman ganti password adalah halaman yang digunakan user untuk mengganti password.



Gambar 4.11 Halaman ganti password

Terdapat data yang harus diisi pengguna untuk mengganti password yaitu password lama, baru, dan konfirmasi password baru.

10. Halaman Riwayat Pekerjaan

Halaman riwayat pekerjaan adalah halaman yang digunakan pengguna untuk melihat riwayat pekerjaan.

10.1. Level PPC

Halaman untuk riwayat pekerjaan PPC



Gambar 4.12 Halaman riwayat pekerjaan level PPC

10.2 Level Supervisor

Halaman riwayat pekerjaan level Supervisor.



Gambar 4.13 Halaman riwayat pekerjaan level Supervisor

Halaman riwayat pekerjaan memiliki fitur untuk filter berdasarkan status.

10.3. Level Worker

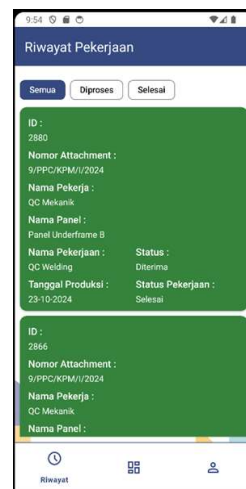
Halaman riwayat pekerjaan Level Worker baik mekanik maupun elektrik.



Gambar 4.14 Halaman riwayat pekerjaan level worker

10.4. Level QC

Halaman riwayat pekerjaan Level QC.



Gambar 4.15 Halaman riwayat pekerjaan level Worker QC

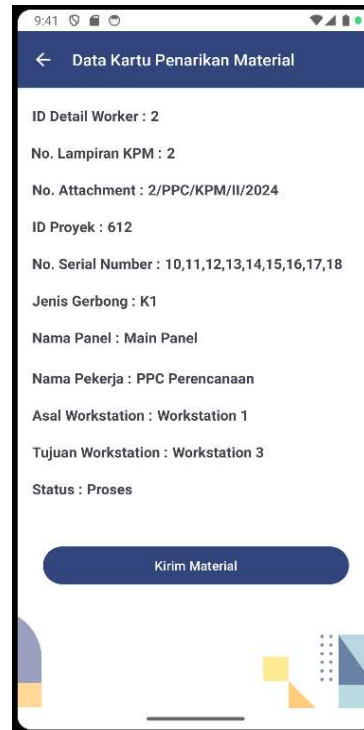
Halaman ini memiliki fitur untuk menyaring data berdasarkan status.

11. Halaman Kerja

Halaman kerja adalah halaman kerja untuk pekerja.

11.1. Level PPC

Halaman kerja level PPC yang digunakan jika ingin mengirim material

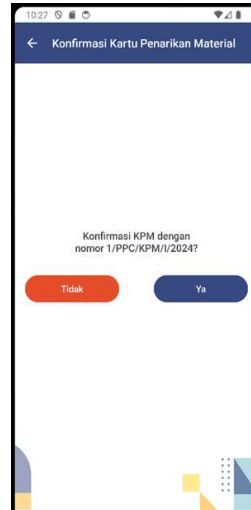


Gambar 4.16 Halaman kerja level Worker PPC

Halaman ini berisi data data seperti nomor attachment, nomor serial, jenis gerbong, nama panel dan lain lain.

11.2 Level Supervisor

Halaman kerja untuk level Supervisor.

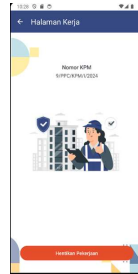


Gambar 4.17 Halaman kerja level supervisor

Halaman ini digunakan untuk Supervisor menerima atau menolak KPM.

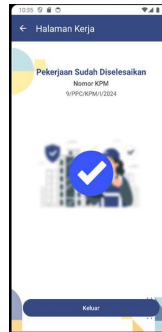
11.3. Level Worker

Halaman kerja worker untuk level worker

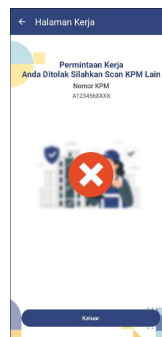


Gambar 4.18 Halaman bekerja level worker

Ketika pengguna menghentikan pekerjaan maka pengguna diminta untuk memotret hasil pekerjaan.



Gambar 4.19 Halaman pekerjaan selesai level worker



Gambar 4.20 Halaman pekerjaan ditolak level worker

Terdapat 3 halaman berbeda ketika pengguna masuk ke halaman kerja tergantung status dari kerja tersebut.

12.4. Level QC

Halaman kerja Quality Control yang digunakan untuk melakukan penjaminan mutu produk.



Gambar 4.21 Halaman bekerja level QC



Gambar 4.21 Halaman catatan tidak lolos level QC

Ketika produk tidak lolos QC maka pengguna mengisi catatan alasan.

b. Cuplikan Kode

Berikut cuplikan kode dari halaman halaman yang ada di aplikasi mobile Rekachain.

1. Halaman Login

Cuplikan halaman login untuk autentikasi pengguna, ketika pengguna sudah login maka aplikasi akan menerima token.

```
override fun onCreate(savedInstanceState: Bundle?) {
    AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_NO)
    super.onCreate(savedInstanceState)
    binding = ActivityAuthLoginBinding.inflate(layoutInflater)
    val view = binding.root
    setContentView(view)

    enableEdgeToEdge()
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
        val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
        insets
    }

    tokenHandler = TokenHandler(context = this)

    initView()

    // Check user login
    if(tokenHandler.getToken() != null) {
        val intent = Intent(packageContext, MainActivity::class.java)
        startActivity(intent)
    }

    val loginRepository = LoginRepository(RetrofitClient.getInstance(context = this, tokenHandler))
    authViewModel = AuthViewModel(loginRepository, tokenHandler, context = this, view = this)

    binding.filledButton.setOnClickListener {
        handleLogin()
    }

    binding.forgotpasswordlink.setOnClickListener {
        val intent = Intent(packageContext, ForgotPasswordActivity::class.java)
        startActivity(intent)
    }

    observeViewModel()
}
```

Gambar 4.22 Cuplikan kode halaman login

token yang didapat dari server digunakan untuk melakukan request ke data data yang dibutuhkan.

2. Halaman Lupa Password

Cuplikan kode Halaman Lupa Password, disini pengguna diminta untuk memasukkan password lama, password baru, dan konfirmasi password baru.

```
class ForgotPasswordActivity: AppCompatActivity() { @Override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState) setContentView(R.layout.activity_forgot_password) binding = ActivityForgotPasswordBinding.inflate(layoutInflater) val view = binding.root toolbar.setSupportActionBar(binding.toolbar) binding.btnBackLogin.setOnClickListener { val intent = Intent(packageContext, LoginActivity::class.java) startActivity(intent) } val toolbar = view.findViewById<Toolbar>(R.id.toolbar) toolbar.setNavigationOnClickListener { val intent = Intent(packageContext, LoginActivity::class.java) startActivity(intent) } }
```

Gambar 4.23 Cuplikan kode halaman lupa password

3. Halaman Dashboard

Cuplikan kode Halaman Dashboard, disini terdapat perbedaan, jika pengguna merupakan Supervisor maka terdapat tombol untuk melihat data request pekerja.

```
class DashboardFragment : Fragment() { private lateinit var binding: FragmentDashboardBinding private lateinit var tokenHandler: TokenHandler private lateinit var tokenHandler: TokenHandler override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState) requestPermissionsLauncher = registerForActivityResult(ActivityResultContracts.RequestPermission() { isGranted: Boolean -> if (isGranted) { Toast.makeText(requireContext(), "Camera permission granted", Toast.LENGTH_LONG).show() setupCamera() // Setup camera or any other camera related setup } else { Toast.makeText(requireContext(), "Camera permission denied", Toast.LENGTH_LONG).show() } }) tokenHandler = TokenHandler(requireContext()) override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View? { binding = FragmentDashboardBinding.inflate(inflater, container, false) cameraPermissions() val loginData = tokenHandler.getLoginResponse() val loginResponse = Gson().fromJson(loginData, LoginResponse::class.java) val levelName = loginResponse.roleLevel if (levelName == LevelFour.SUPERVISOR_Level) { binding.btnRequest.visibility = View.VISIBLE binding.btnRequest.setOnClickListener { workRequest() } } }
```

Gambar 4.24 Cuplikan kode halaman dashboard

Jika bukan Supervisor maka hanya ada tombol Scan yang ada di Halaman Dashboard.

4. Halaman Data Kartu Penarikan Material

Cuplikan Halaman Data KPM (Kartu Penarikan Material) terdapat 2 jenis KPM yaitu KPM untuk pekerja mekanik dan elektrik dan KPM untuk pekerja Assembly.

```
override fun onCreate(savedInstanceState: Bundle?) {
    val intent = Intent(packageContext, LoginActivity::class.java)
    startActivity(intent)
}

val resultId: String? = intent.getStringExtra("resultId")
val rawValue: String? = intent.getStringExtra("result")
tokenHandler = TokenHandler(context, this)
val loginData = tokenHandler.getLoginResponse()
val loginResponse = Gson().fromJson(loginData, LoginResponse::class.java)
val levelName = loginResponse.role.level
val divisionName = loginResponse.division?.name
var serialNumberId: String? = null
var attachmentNumber: String? = null

if (levelName == LevelEnum.PPC.level) {
    if (rawValue.isNullOrEmpty()) {
        this.finish()
    } else {
        binding.btnNext.text = "Kirin Material"
        val isConstainSNData = containsSNData(rawValue)

        if (isConstainSNData) {
            getAttachment()
        } else {
            getTrainsetAttachment()
        }
    }
} else {
    // set route scanner based role & division
    if (divisionName == DivisionEnum.MEKANIK.division || divisionName == DivisionEnum.ELEKTRIK.division) {
        getTrainsetAttachment()
    } else if (divisionName == DivisionEnum.ASSEMBLY.division) {
        if (levelName == LevelEnum.SUPERVISOR.level) {
            getAttachment() //scan KPM
        } else if (levelName == LevelEnum.WORKER.level || levelName == LevelEnum.QC.level) {
            getAttachmentDetail() //scan KPM-SN
        }
    }
}
}
```

Gambar 4.25 Cuplikan kode halaman data kartu penarikan material

Halaman ini juga berfungsi sebagai halaman untuk PPC ketika ingin mengirim material.

5. Halaman Data Request Pekerja

Cuplikan kode Data Request pekerja, di halaman ini Supervisor dapat melihat data request pekerja dan Supervisor dapat menerima maupun menolak request pekerja.

```
override fun onCreate(savedInstanceState: Bundle?) {
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
        val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
    }
    insets

    chipGroup = BindingChipGroup()
    rvWorkRequest = findViewById<RecyclerView>(R.id.cv_work_request)
    rvWorkRequest.layoutManager = LinearLayoutManager(context, this)
    rvWorkRequest.setHasFixedSize(true)

    tokenHandler = TokenHandler(context, this)
    if(tokenHandler.getToken().isNullOrEmpty() || tokenHandler.getLoginResponse().isNullOrEmpty()) {
        val intent = Intent(context, LoginActivity::class.java)
        startActivity(intent)
    }

    initView()

    val loginData = tokenHandler.getLoginResponse()
    val loginResponse = gson().fromJson(loginData, LoginResponse::class.java)

    if (loginResponse != null) {
        level = loginResponse.role.level
        division = loginResponse.division?.name?.toString()
    }

    val attachmentWorkerRepository = AttachmentWorkerRepository(RetrofitClient.getInstance(context, tokenHandler))
    workRequestViewModel = WorkRequestViewModel(attachmentWorkerRepository, view, this, context, this)

    if(division == DivisionEnum.ASSFRBY.division) {
        listWorkRequestSpvAssyAdapter = ListWorkRequestSpvAssyAdapter()
        rvWorkRequest.adapter = listWorkRequestSpvAssyAdapter
        listWorkRequestSpvAssyAdapter.listener = this
    } else if (division == DivisionEnum.REKAWTK.division || division == DivisionEnum.FLEKTRIK.division) {
        listWorkRequestSpvRekAdapter = ListWorkRequestSpvRekAdapter()
        rvWorkRequest.adapter = listWorkRequestSpvRekAdapter
        listWorkRequestSpvRekAdapter.listener = this
    } else {
        finish()
    }
}
```

Gambar 4.26 Cuplikan kode halaman data request pekerja

Halaman Data Request Pekerja memiliki fitur untuk menyaring data berdasarkan status.

6. Halaman Konfirmasi Pekerja

Halaman digunakan Supervisor dalam memilih apakah menerima atau menolak request pekerja.

```
class ConfirmWorkRequestActivity : AppCompatActivity(), BaseApiView {
    private fun observeViewModel() {
        workRequestViewModel.actionWorkRequestResult.observe(owner = this){result ->

            if(res?.acceptanceStatus == AcceptanceStatusEnum.ACCEPTED.status){
                showInfoDialog(title = "Berhasil", message = "Berhasil menugaskan pekerja", isActivityFinish: true)
            }else {
                showInfoDialog(title = "Berhasil", message = "Berhasil Menolak pekerja", isActivityFinish: true)
            }
        }else{
            dismissLoading()
            showInfoDialog(title = "Gagal", message = "Error", isActivityFinish: true)
        }
    }
}

private fun btnAction() {
    val id = intent.extras?.getInt(key = "id")
    val name = intent.extras?.getString(key = "name")
    val step = intent.extras?.getString(key = "step")

    binding.tvTitle.text = "Tugaskan {name} untuk melakukan {step}"

    if (id != null) {
        binding.btnConfirm.setOnClickListener {
            if (division == DivisionEnum.ELEKTRIK.division || division == DivisionEnum.MEKANIK.division) {
                workRequestViewModel.actionWorkRequestSpvMKEK(id, AcceptanceStatusEnum.ACCEPTED.status)
            } else if (division == DivisionEnum.ASSEMBLY.division) {
                workRequestViewModel.actionWorkRequestSpvAssy(id, AcceptanceStatusEnum.ACCEPTED.status)
            }
        }
        binding.btnReject.setOnClickListener {
            if (division == DivisionEnum.ELEKTRIK.division || division == DivisionEnum.MEKANIK.division) {
                workRequestViewModel.actionWorkRequestSpvMKEK(id, AcceptanceStatusEnum.DECLINED.status)
            } else if (division == DivisionEnum.ASSEMBLY.division) {
                workRequestViewModel.actionWorkRequestSpvAssy(id, AcceptanceStatusEnum.DECLINED.status)
            }
        }
    } else {
        finish()
    }
}
```

Gambar 4.27 Cuplikan kode halaman konfirmasi pekerja

Ketika Supervisor menolak pekerjaan worker maka akan merubah status diterima tidaknya sebuah pekerjaan.

7. Halaman Profil

Cuplikan kode halaman profil, di halaman ini terdapat tiga tombol yaitu tombol untuk melihat data detail akun, tombol untuk berpindah ke halaman ganti password, dan tombol untuk Logout.

```
class ProfileFragment : Fragment(), BaseApiView {
    override fun onCreateView(view: View, savedInstanceState: Bundle?) {
        super.onCreateView(view, savedInstanceState)

        val loginData = tokenHandler.getLoginResponse()
        loginResponse = Gson().fromJson(loginData, LoginResponse::class.java)

        binding.tvName.setText(loginResponse.user.name)
        binding.tvNip.setText(loginResponse.user.nip)

        initView()

        if(!loginResponse.user.image_path.isNullOrEmpty()){
            try {
                Glide.with(fragment = this)
                    .load("http://$BuildConfig.STORAGE_URL/${loginResponse.user.image_path}")
                    .error(R.drawable.ic_user_profile)
                    .into(binding.ivProfile)
            } catch (e: Exception) {
                val imgResId = R.drawable.ic_user_profile
                val resId = imgResId
                binding.ivProfile.setImageResource(resId)
                Toast.makeText(requireContext(), "Gagal memuat gambar profil", Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```

Gambar 4.28 Cuplikan kode halaman profil

8. Halaman Akun

Cuplikan kode halaman akun, halaman ini digunakan untuk melihat lebih detail data pengguna.

```
class AccountActivity: AppCompatActivity(), BaseApiView {
    private lateinit var binding: ActivityAccountBinding
    private lateinit var tokenHandler: TokenHandler
    private lateinit var loginResponse: LoginResponse
    var progressDialog: Dialog? = null
    private lateinit var profileViewModel: ProfileViewModel
    private var path:String = ""

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE_NIGHT_NO)
        binding = ActivityAccountBinding.inflate(layoutInflater)
        tokenHandler = TokenHandler(context = this)
        val profileRepository = ProfileRepository(RetrofitClient.getInstance(context = this, tokenHandler))

        val view = binding.root
        setContentView(view)

        if(tokenHandler.getToken().isNullOrEmpty() || tokenHandler.getLoginResponse().isNullOrEmpty()) {
            val intent = Intent(context = this, LoginActivity::class.java)
            startActivity(intent)
        }

        val user = tokenHandler.getLoginResponse()
        loginResponse = Gson().fromJson(user, LoginResponse::class.java)

        profileViewModel = ProfileViewModel(profileRepository, tokenHandler, context = this, view = this)

        binding.tvName.setText(loginResponse.user.name)
        binding.tvNip.setText(loginResponse.user.nip)
        binding.tvEmail.setText(loginResponse.user.email)
        binding.tvPhoneNum.setText(loginResponse.user.phone_number)
        binding.tvJabatan.setText(loginResponse.role.name)
    }
}
```

Gambar 4.29 Cuplikan kode halaman akun

Halaman akun memiliki fitur untuk mengubah foto profil dengan mengklik foto profil.

11. Halaman Kerja

11.1. Supervisor

Cuplikan kode Supervisor, di halaman ini supervisor dapat menerima atau menolak KPM (Kartu Penarikan Material) tergantung kondisi.

```
class KPMConfirmFragment : Fragment(), BaseAPIView {
    private lateinit var binding: FragmentKPMConfirmBinding
    private lateinit var viewModel: WorkingViewModel
    private lateinit var tokenHandler: TokenHandler
    var progressDialog: Dialog? = null

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        binding = FragmentKPMConfirmBinding.inflate(inflater)
        initView()
        tokenHandler = TokenHandler(requireContext())

        if (tokenHandler.getToken().isNullOrEmpty() || tokenHandler.getLoginResponse().isNullOrEmpty()) {
            val intent = Intent(requireContext(), LoginActivity::class.java)
            startActivity(intent)
        }

        val attachmentWorkerRepository = AttachmentWorkerRepository(RetrofitClient.getInstance(requireContext(), tokenHandler))
        viewModel = WorkingViewModel(attachmentWorkerRepository, view = this, requireContext())

        val resultId = arguments?.getString(key = "RESULT_ID")
        val attachmentNumber = arguments?.getString(key = "ATTACHMENT_NUMBER")
        binding.tvTitle.text = "Konfirmasi KPM dengan nomor {attachmentNumber}"
    }
}
```

Gambar 4.32.1 Cuplikan kode halaman kerja Supervisor

Jika Supervisor menolak KPM maka supervisor diminta memberikan catatan.

11.2 Worker

Cuplikan kode Halaman Kerja Worker, halaman ini digunakan untuk mengkonfirmasi jika pekerjaan sudah selesai.

```
class WorkingFragment : Fragment() {
    private lateinit var binding: FragmentWorkingBinding
    private var handler: Handler = Handler()
    private var runnable: Runnable? = null
    private var delay = 1000 * 60 * 5
    private lateinit var tokenHandler: TokenHandler
    private lateinit var historyViewModel: HistoryViewModel
    private var resultId: String? = null
    private var attachmentNumber: String? = null

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        binding = FragmentWorkingBinding.inflate(inflater)

        tokenHandler = TokenHandler(requireContext())

        val attachmentWorkerRepository = AttachmentWorkerRepository(RetrofitClient.getInstance(requireContext(), tokenHandler))
        historyViewModel = HistoryViewModel(attachmentWorkerRepository, tokenHandler, requireContext())

        resultId = arguments?.getString(key: "RESULT_ID")
        attachmentNumber = arguments?.getString(key: "ATTACHMENT_NUMBER")

        if (resultId != null && attachmentNumber != null) {
            binding.tvAttachmentNum.text = attachmentNumber
        } else {
            Toast.makeText(requireContext(), "Nomor Attachment tidak ditemukan", Toast.LENGTH_SHORT).show()
            activity?.finish()
        }

        binding.toolbar.setNavigationOnClickListener {
            activity?.finish()
        }

        binding.btnStop.setOnClickListener {
            ImagePicker.with(fragment: this).cameraOnly().compress(maxSize: 2000).start()
        }
    }
}
```

Gambar 4.33.2 Cuplikan kode halaman kerja Worker

Jika pengguna mengklik tombol hentikan pekerjaan maka pengguna diminta memotret hasil kerjanya.

11.3. Quality Control

Cuplikan kode Quality Control, pada halaman ini pengguna Quality Control dapat menerima atau menolak produk.

```
class ConfirmQcFragment : Fragment(), BaseApplyView {
    private lateinit var binding: FragmentConfirmQcBinding
    private lateinit var viewModel: WorkingViewMode
    private lateinit var historyViewModel: HistoryViewMode
    private lateinit var tokenHandler: TokenHandler
    var progressDialog: Dialog? = null
    var handler: Handler? = null
    var runnable: Runnable? = null
    var delay = 1000 * 60 * 5
    private var resultId: String? = null
    private var serialNumber: String? = null
    private var attachmentNumber: String? = null

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        binding = FragmentConfirmQcBinding.inflate(inflater)
        viewModel = FragmentConfirmQcBinding.inflate(inflater)
        initView()
        tokenHandler = TokenHandler(requireContext())
        if (tokenHandler.getToken().isNullOrEmpty() || tokenHandler.getLoginResponse().isNullOrEmpty()) {
            val intent = Intent(requireContext(), LoginActivity::class.java)
            startActivity(intent)
        }
        val user = tokenHandler.getLoginResponse()
        val loginResponse = user?.fromJson<User, LoginResponse>(class.java)
        val attachmentWorkerRepository = AttachmentWorkerRepository(RetrofitClient.getInstance(requireContext()), tokenHandler)
        viewModel = WorkingViewMode(attachmentWorkerRepository, this, requireContext())
        historyViewModel = HistoryViewMode(attachmentWorkerRepository, tokenHandler, requireContext())
    }
}
```

Gambar 4.34.3 Cuplikan kode halaman Quality Control

Jika pengguna Quality Control menolak produk maka pengguna diminta memasukkan catatan dan memasukan berapa jumlah yang tidak lolos.

4.2 Implementasi sistem di industri

Berikut merupakan dokumentasi pengujian ke pihak industri.



Gambar 4.35 Dokumentasi

Pengujian ini dilakukan secara offline di PT. Rekindo Global Jasa di Jl. Candi Sewu No.30, Madiun Lor, Kec. Manguharjo, Kota Madiun, Jawa Timur 63122.

4.3 Hasil yang didapat selama magang

Beberapa hal yang didapatkan selama magang :

1. Pengalaman praktis
2. Skill teknis
3. Kerjasama tim
4. Menambah portofolio

BAB V PENUTUP

5.1. Kesimpulan

Berdasarkan kegiatan yang telah dilaksanakan selama periode magang, dapat disimpulkan bahwa program magang ini dampak positif seperti belajar berkolaborasi secara tim, menambah pengalaman dalam berkomunikasi, menambah portfolio, menambah skill teknis.

Secara umum, program magang ini tidak hanya membantu penulis meningkatkan kemampuan Teknik, tetap juga memberikan bekal yang berguna untuk bekerja dengan orang lain atau secara tim, oleh karena itu penulis berharap agar pengalaman ini dapat menjadi fondasi untuk bekerja di dunia nyata.

5.2. Saran

Saran penulis kepada tempat magang industri yaitu menyediakan sesi evaluasi rutin untuk membahas kemajuan tim , dan mencari Solusi untuk kendali yang dihadapi, karena selama magang ini dilaksanakan sesi evaluasi jarang dilakukan.

REFERENSI

LAMPIRAN A. TERM OF REFERENCE (TOR)

1. Latar Belakang

Hidup dalam era perkembangan teknologi informasi saat ini, memberikan kemudahan dalam berbagai bidang. Sejalan dengan hal itu diperlukan sumber daya manusia (SDM) yang mampu mendayagunakan kemampuannya di segala bidang khususnya di bidang teknologi informasi dalam dunia kerja yang sebenarnya. Mahasiswa sebagai salah satu anggota masyarakat akademis di lingkungan Politeknik diharapkan mampu menjadi lulusan yang terampil, profesional, dan mempunyai etos kerja yang tinggi. Politeknik Negeri Malang sebagai salah satu politeknik di Indonesia selalu menyesuaikan kurikulum pendidikannya sesuai dengan perkembangan dunia kerja yang sesungguhnya. Oleh karena itu, Politeknik Negeri Malang diharapkan mampu mencetak lulusan yang berkualitas dan mampu mengimplementasikan ilmu yang diperoleh saat mengikuti perkuliahan sebagai pondasi dalam menghadapi dan menyelesaikan berbagai permasalahan yang terjadi pada dunia kerja.

Permasalahan yang terjadi dalam dunia kerja berbeda dengan permasalahan saat perkuliahan. Dengan situasi tersebut, mahasiswa D-IV Sistem Informasi Bisnis dituntut untuk mengikuti dan beradaptasi dengan lingkungan pekerjaan sehingga setelah memasuki dunia kerja telah memiliki bekal yang cukup dan siap menghadapi pekerjaan.

Dalam usaha untuk menyesuaikan dan mengikuti perkembangan industri, kegiatan magang industri dinilai sebagai sarana yang efektif untuk menyesuaikan diri dengan perkembangan di dunia kerja. Oleh karena itu mahasiswa hendaknya dapat memanfaatkan kegiatan ini sebaik-baiknya.

Menimbang hal-hal tersebut diatas, maka penulis memilih PT. Rekaindo Global Jasa sebagai wadah serta tempat untuk melakukan magang industri.

2. Tujuan Kegiatan

Magang industri ini bertujuan untuk mempersiapkan mahasiswa menjadi tenaga profesional, disiplin, kreatif, dan jujur untuk meningkatkan etos kerja

3. Jadwal dan Tempat Pelaksanaan

Tanggal : 15 Juli 2024 - 15 Agustus 2024

Jam : 07.30-selesai

Mitra : PT. Rekaindo Global Jasa

Tempat : Online

4. Peserta

Jumlah peserta kegiatan magang industri yang berasal dari Politeknik Negeri Malang adalah 8 orang mahasiswa Jurusan Teknologi Informasi dengan Program Studi Sistem Informasi Bisnis.

5. Penutup

Program magang ini diharapkan dapat memberikan manfaat baik bagi peserta maupun perusahaan. Peserta mendapatkan pengalaman dan pengetahuan baru, sementara perusahaan dapat memanfaatkan kontribusi peserta dalam pengembangan aplikasi

LAMPIRAN B. LOG ACTIVITY

LAMPIRAN C. DOKUMEN TEKNIK

A. Deskripsi Kebutuhan

1. Kebutuhan Fungsional

Layanan sistem yang harus disediakan

1. Pengguna dapat melakukan pembuatan proyek baru
2. Pengguna dapat melakukan pembuatan trainset baru dari proyek yang ada
3. Pengguna dapat memilih gerbong di setiap trainset yang ada
4. Pengguna dapat memilih jumlah dan jenis panel di setiap gerbong ada
5. Sistem dapat mencetak laporan ke format PDF

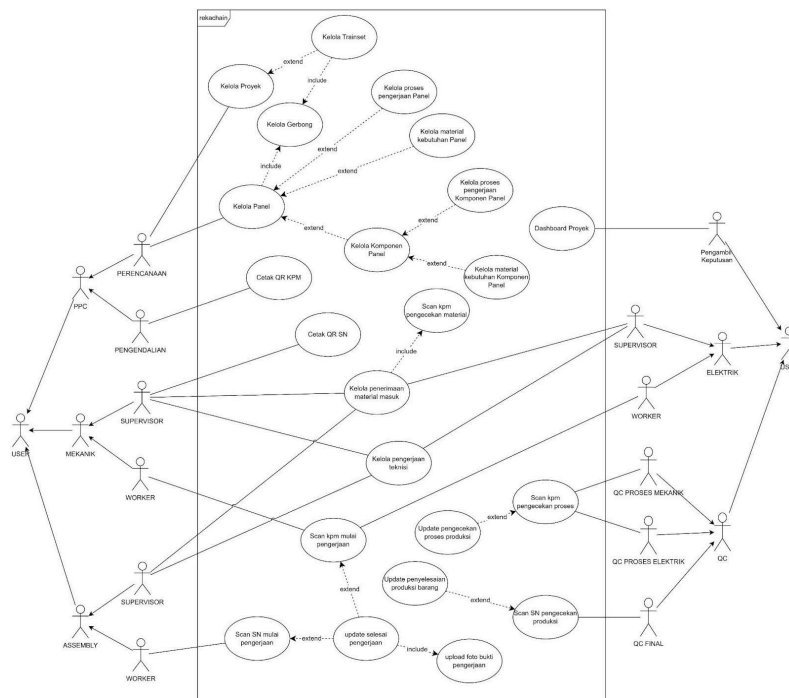
2. Kebutuhan Non Fungsional

Berikut adalah uraian singkat mengenai kebutuhan non fungsional dalam bentuk tabel :

SRS-ID	Kebutuhan	Deskripsi
1	Availability	Aplikasi harus selalu tersedia dan bisa digunakan kapan saja, 24 jam sehari, 7 hari seminggu.
2	Reliability	Aplikasi harus sangat andal dan hanya boleh mengalami kegagalan maksimal 2% dari waktu operasinya.
3	Ergonomy	Antarmuka pengguna harus mudah digunakan dan dipahami, sehingga pengguna baru bisa belajar dalam waktu kurang dari 4 jam.
4	Portability	Aplikasi harus bisa dijalankan di berbagai sistem operasi yang memiliki web browser untuk desktop dan AndroidOS untuk mobile.

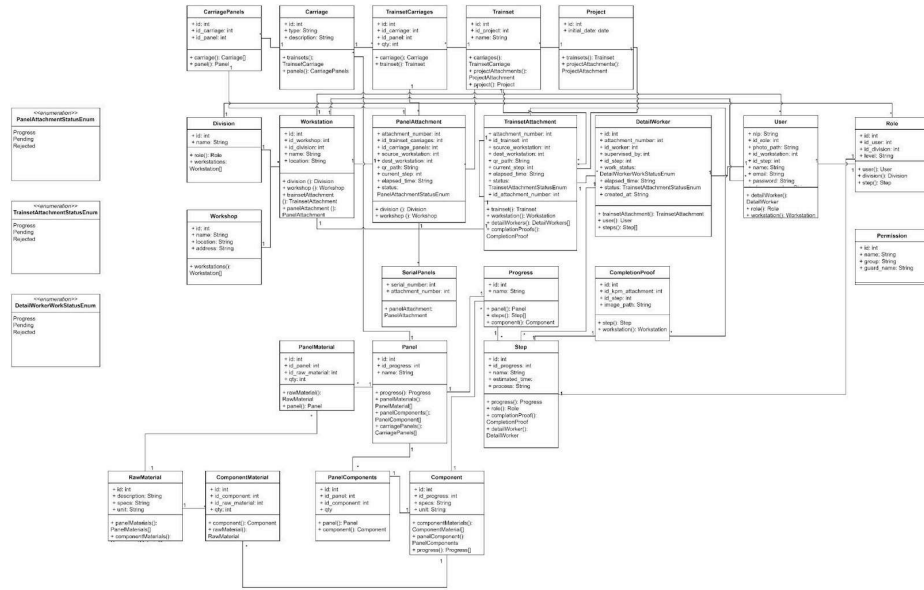
5	Memory	Aplikasi tidak boleh menggunakan lebih dari 512MB memori pada komputer standar.
6	Response time	Aplikasi harus bisa memberikan respon atau hasil dalam waktu kurang dari 5 detik untuk sebagian besar permintaan pengguna.
7	Safety	Aplikasi harus mematuhi standar keselamatan yang berlaku di industri, seperti ISO 26262.
8	Security	Aplikasi harus melindungi data pengguna dengan enkripsi dan mematuhi peraturan perlindungan data.

B. Use Case Diagram



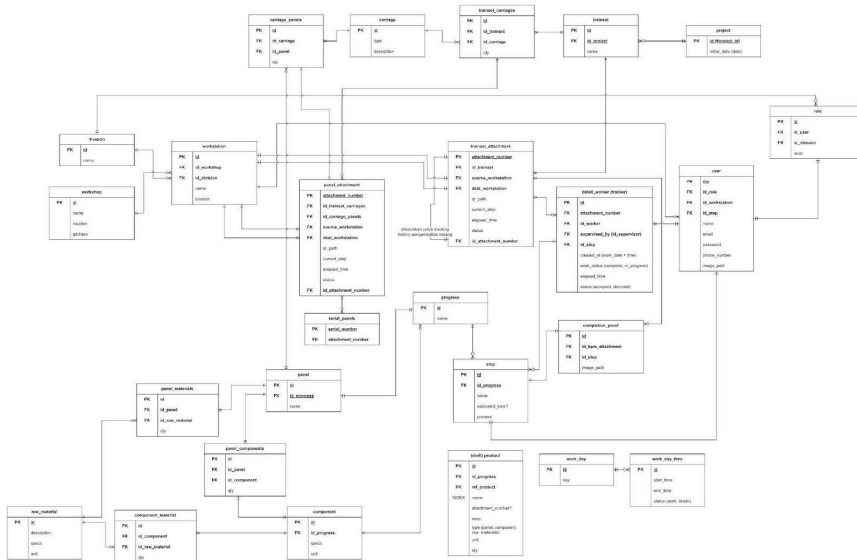
Gambar C.1 Use Case Diagram

C. Class Diagram



Gambar C.2 Class Diagram

D. Entity Relationship Diagram



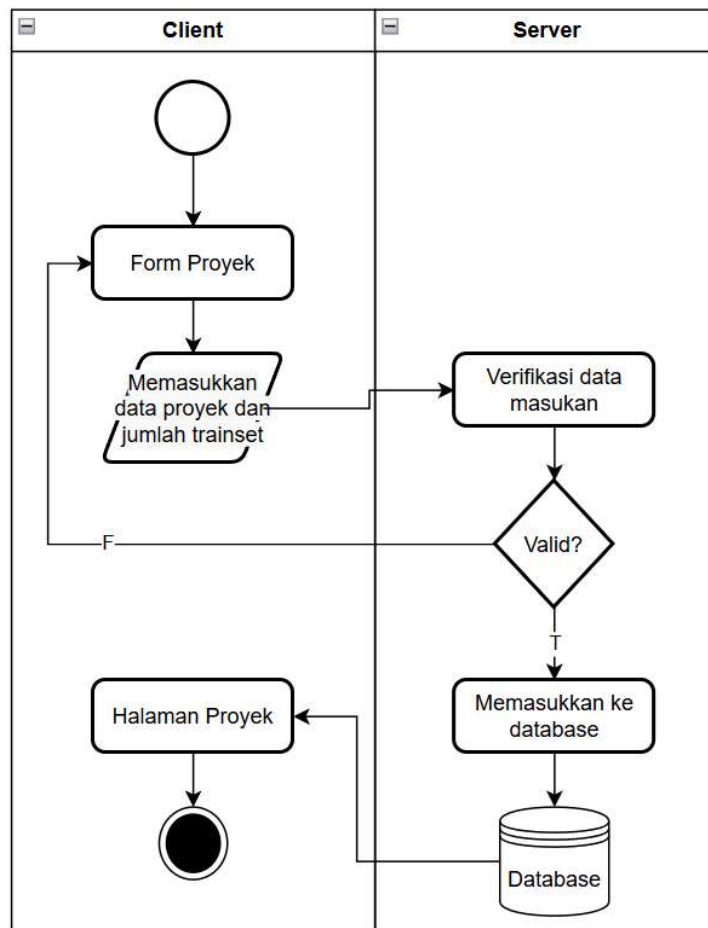
Gambar C.3 Entity Relationship Diagram

E. Entity Relationship Diagram

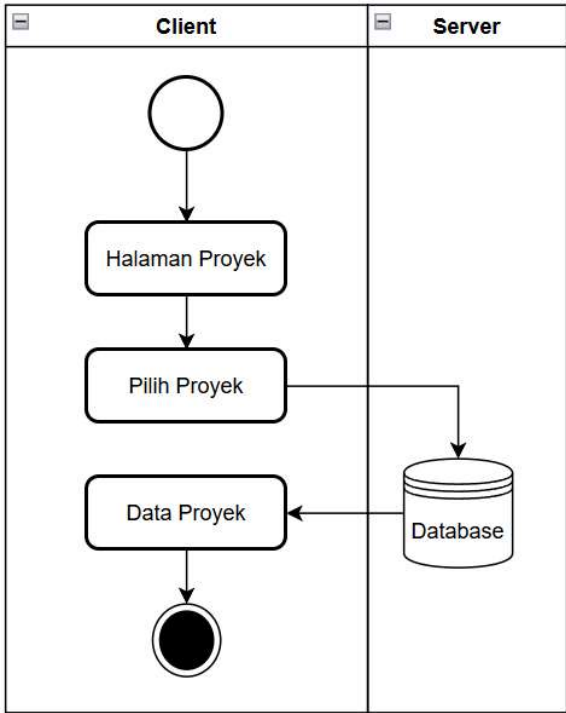
F. Activity Diagram

1. Proses Kelola Proyek

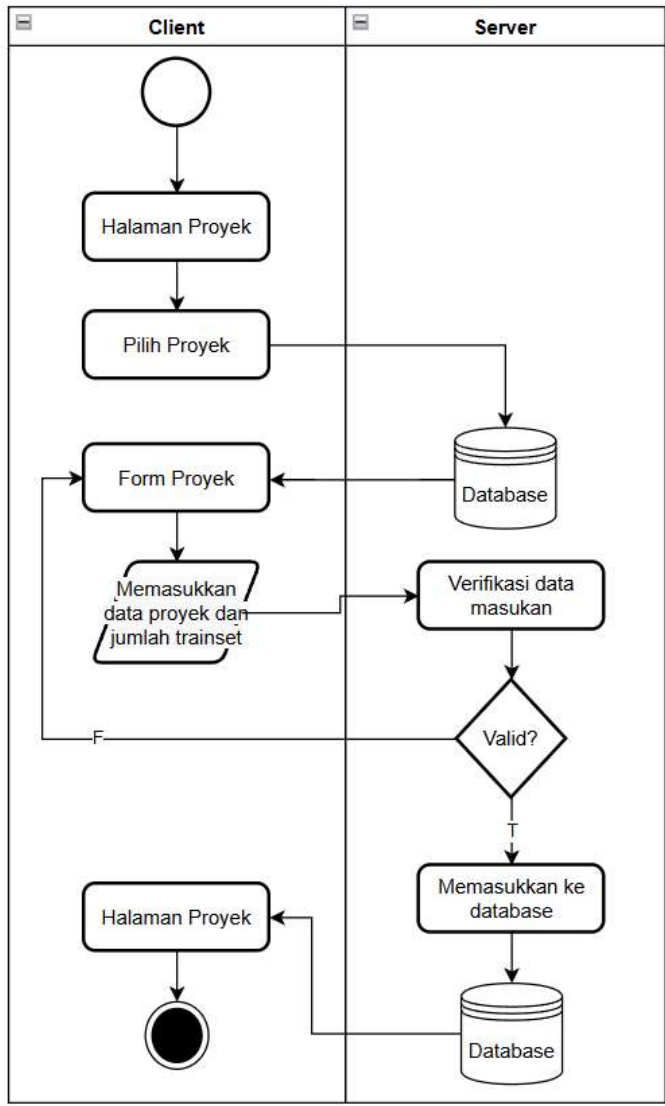
PSPEC 1.1 Buat Proyek
User melakukan pembuatan proyek ke dalam sistem
Masukan : tgl Proyek, nama proyek, jumlah trainset
Keluaran : data proyek dan form trainset



PSPEC 1.2 Cek Proyek
User melakukan pengecekan proyek dari sistem
Masukan : id_proyek
Keluaran : data proyek



<p>PSPEC 1.3 Edit Proyek</p> <p>User melakukan pengeditan proyek dari sistem</p> <p>Masukan : id_proyek, tgl Proyek, nama proyek, jumlah trainset</p> <p>Keluaran : data proyek dan form trainset</p>



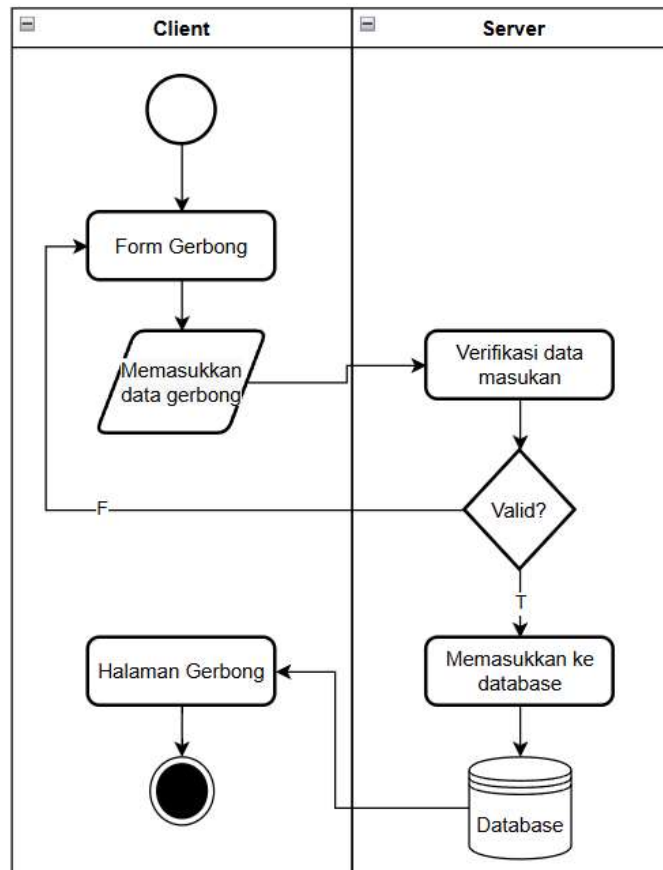
2. Proses Kelola Gerbong

PSPEC 2.1 Buat Gerbong

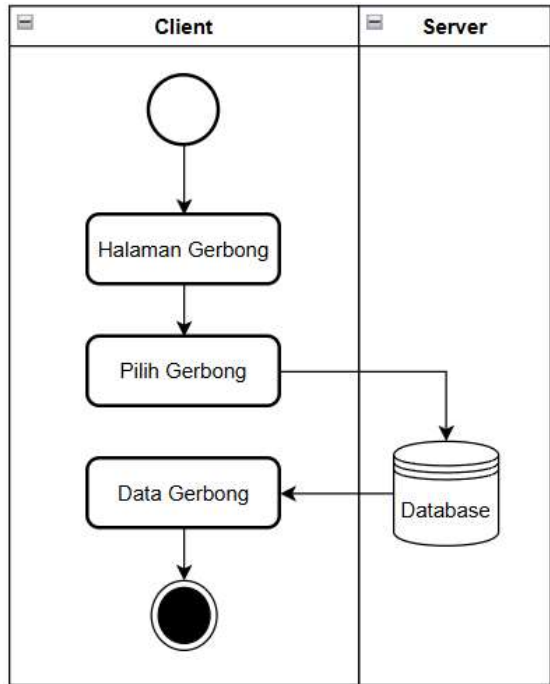
User melakukan pembuatan gerbong ke dalam sistem

Masukan : nama gerbong, tipe gerbong

Keluaran : data gerbong



PSPEC 2.2 Cek Gerbong
User melakukan pengecekan gerbong dari sistem
Masukan : id_gerbong
Keluaran : data gerbong

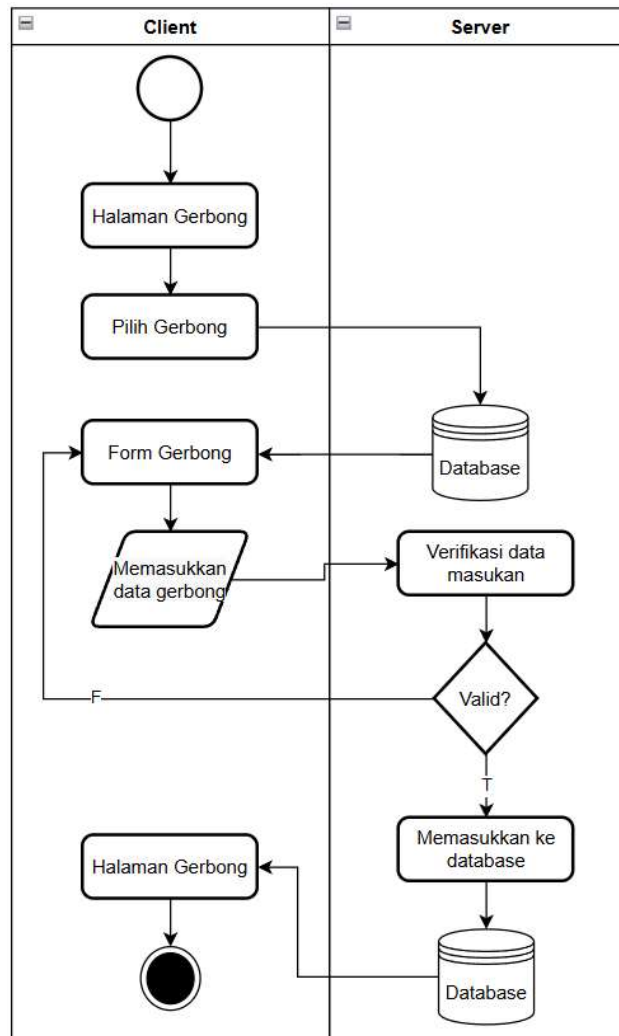


PSPEC 2.3 Edit Gerbong

User melakukan pengeditan gerbong dari sistem

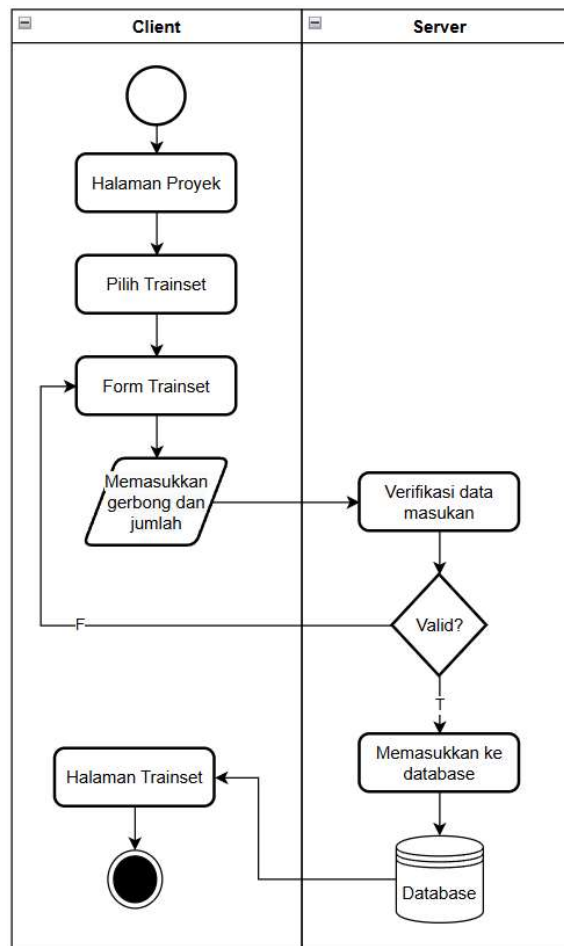
Masukan : id_gerbong, nama gerbong, tipe gerbong

Keluaran : data gerbong

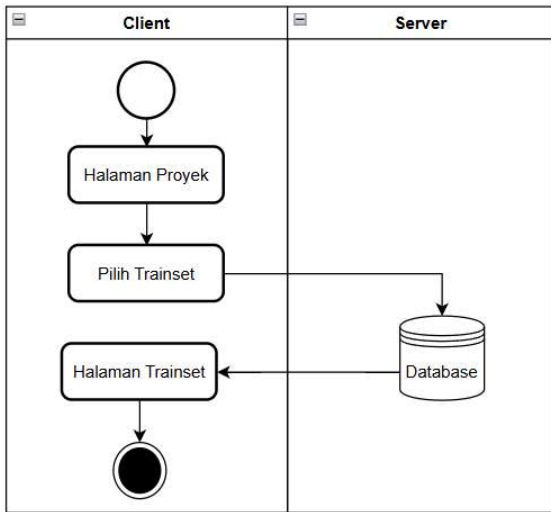


3. Proses Kelola Trainset

PSPEC 3.1 Input/Edit Trainset
User melakukan pembuatan/ pengeditan trainset ke dalam sistem
Masukan :id_trainset, gerbong, jumlah gerbong
Keluaran : data trainset

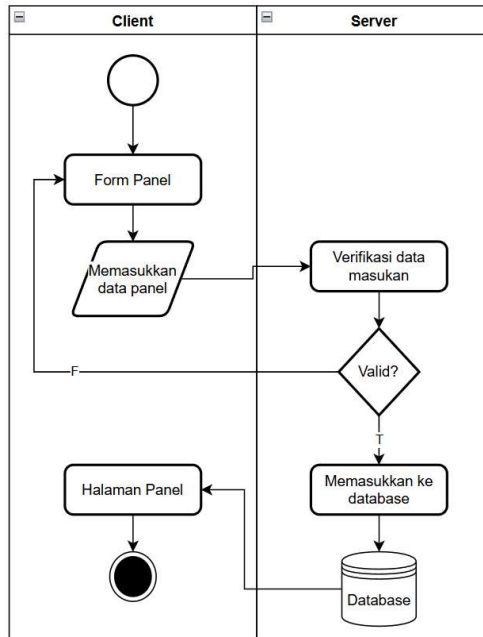


PSPEC 3.2 Cek Trainset
User melakukan pengecekan trainset dari sistem
Masukan : id_trainset
Keluaran : data trainset

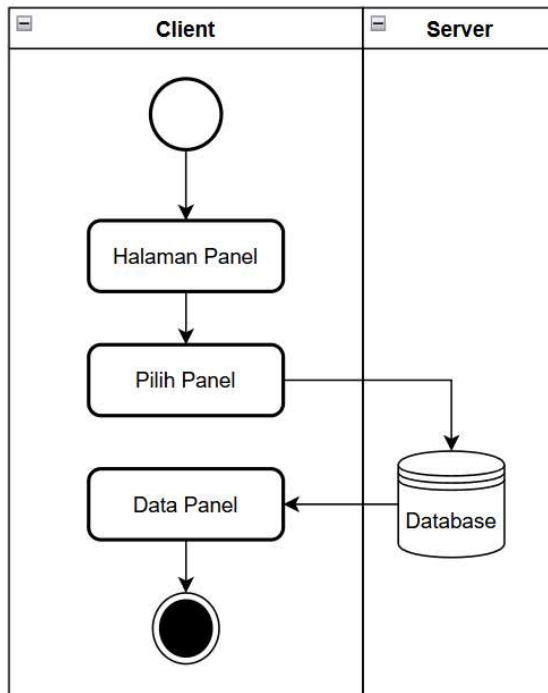


4. Proses Kelola Panel

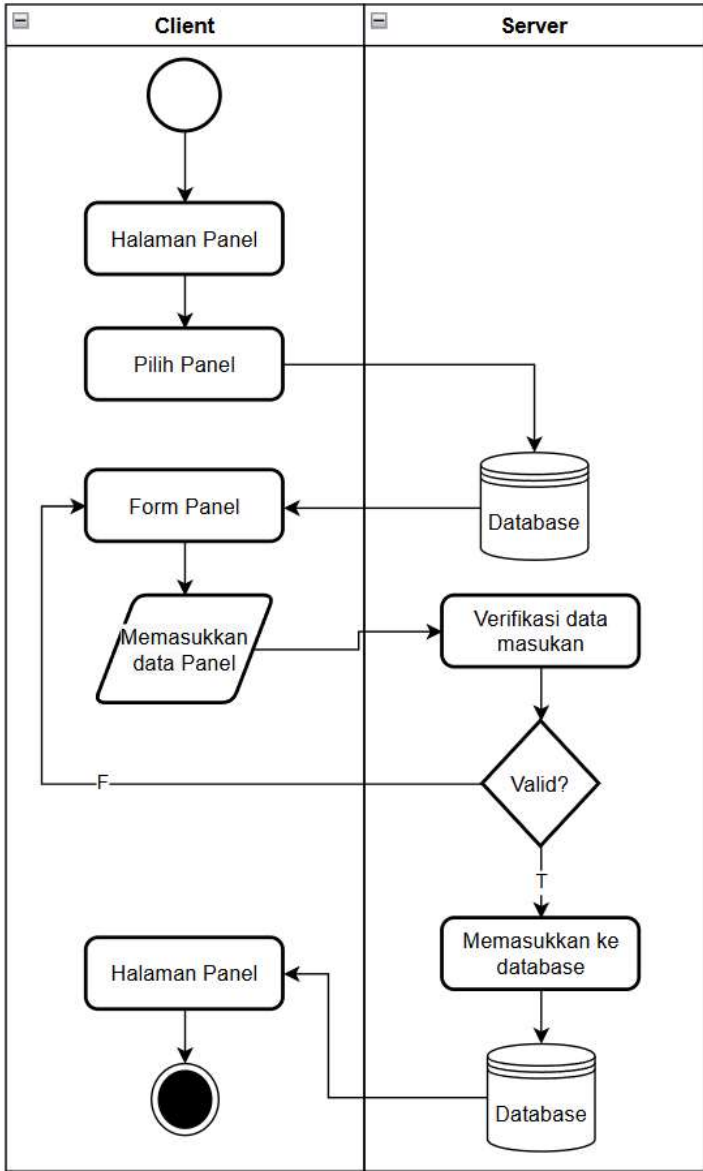
PSPEC 4.1 Input Panel
User melakukan pembuatan panel ke dalam sistem
Masukan : nama panel, gerbong
Keluaran : data panel



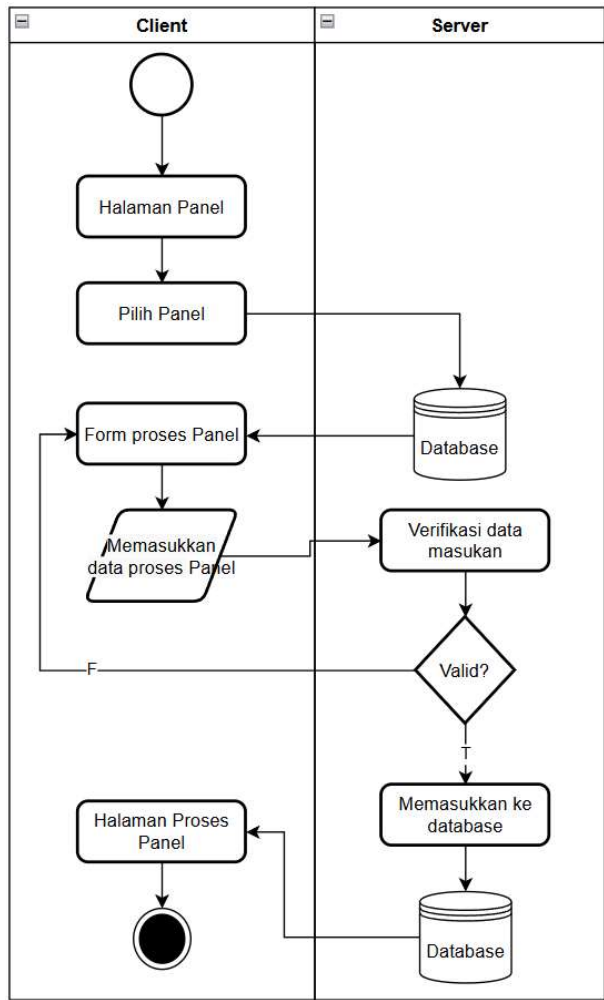
PSPEC 4.2 Cek Panel
User melakukan pengecekan panel dari sistem
Masukan : id_panel
Keluaran : data panel



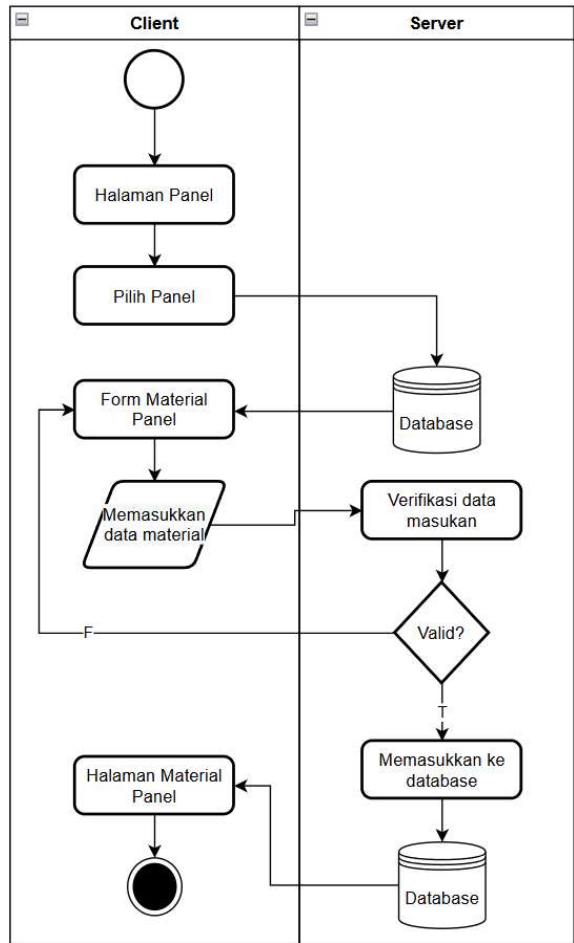
<p>PSPEC 4.3 Edit Panel</p> <p>User melakukan pengeditan panel dari sistem</p> <p>Masukan : id_panel, nama panel, gerbong</p> <p>Keluaran : data panel</p>



<p>PSPEC 4.4 Proses Pengerjaan Panel</p> <p>User melakukan pengeditan proses pengerjaan panel dari sistem</p> <p>Masukan : id_panel, proses, step</p> <p>Keluaran : data panel</p>



<p>PSPEC 4.5 Material Panel</p> <p>User melakukan pengeditan material kebutuhan panel dari sistem</p> <p>Masukan : id_panel, material, jumlah</p> <p>Keluaran : data panel</p>



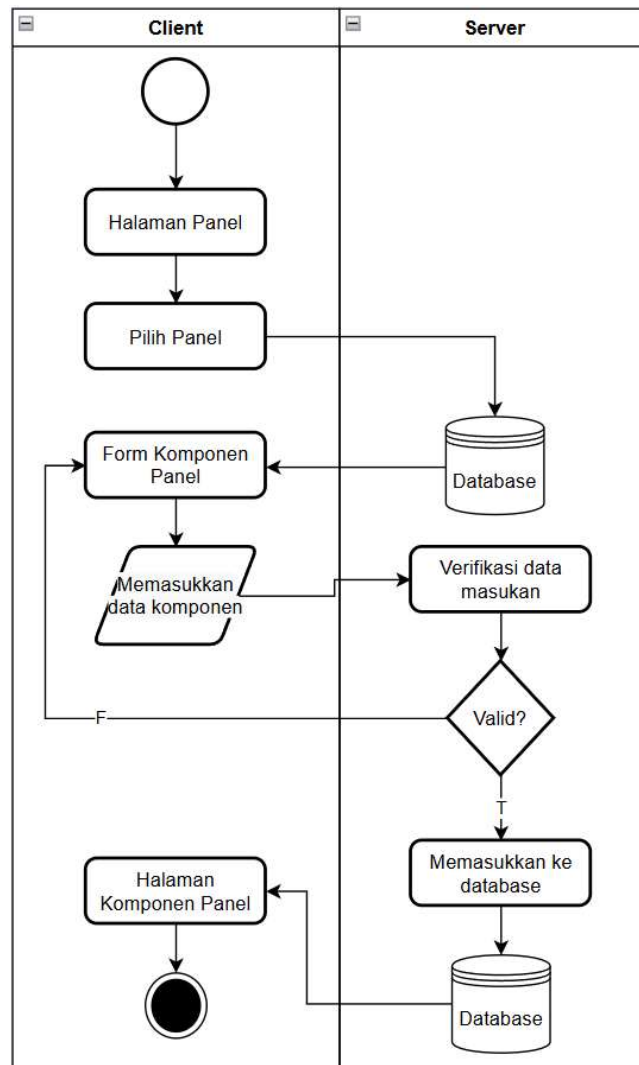
5. Proses Kelola Komponen Panel

PSPEC 5.1 Komponen Panel

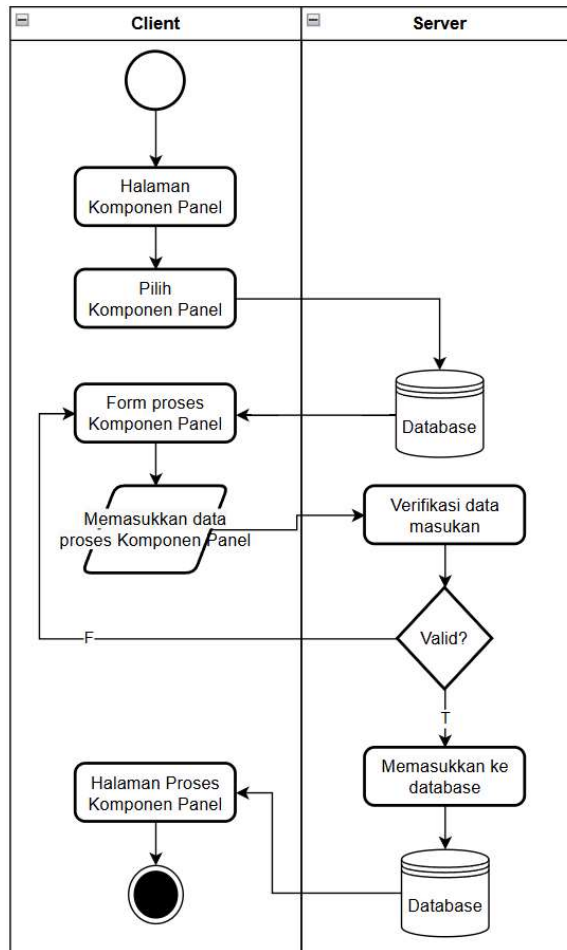
User melakukan pengeditan komponen panel di dalam sistem

Masukan : id_komponen, nama komponen

Keluaran : data komponen



<p>PSPEC 5.2 Proses Pengerjaan Komponen Panel</p> <p>User melakukan pengeditan proses pengerjaan panel dari sistem</p> <p>Masukan : id_komponen, proses, step</p> <p>Keluaran : data komponen</p>

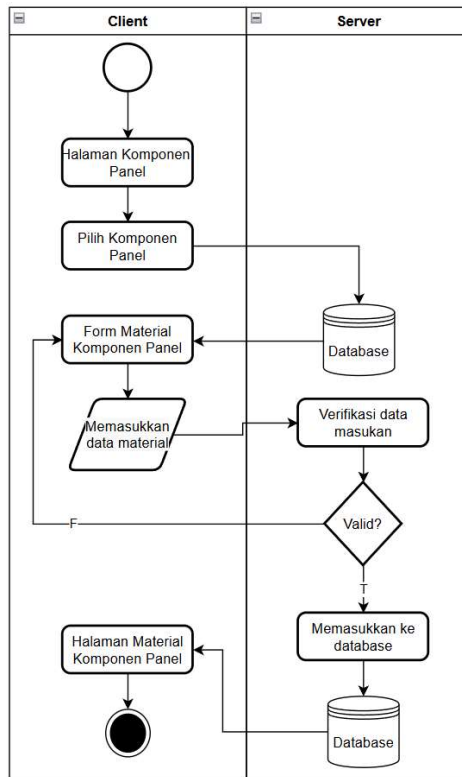


PSPEC 5.3 Material Komponen Panel

User melakukan pengeditan material komponen panel dari sistem

Masukan : id_komponen, material, jumlah

Keluaran : data komponen



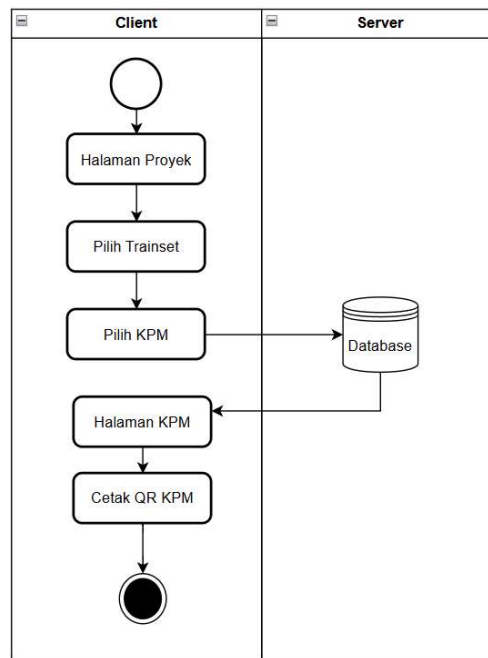
6. Proses Cetak QR

PSPEC 6.1 Cetak QR KPM

User melakukan pencetakan QR Kartu Penarikan Material dari sistem

Masukan : proyek, trainset

Keluaran : data KPM

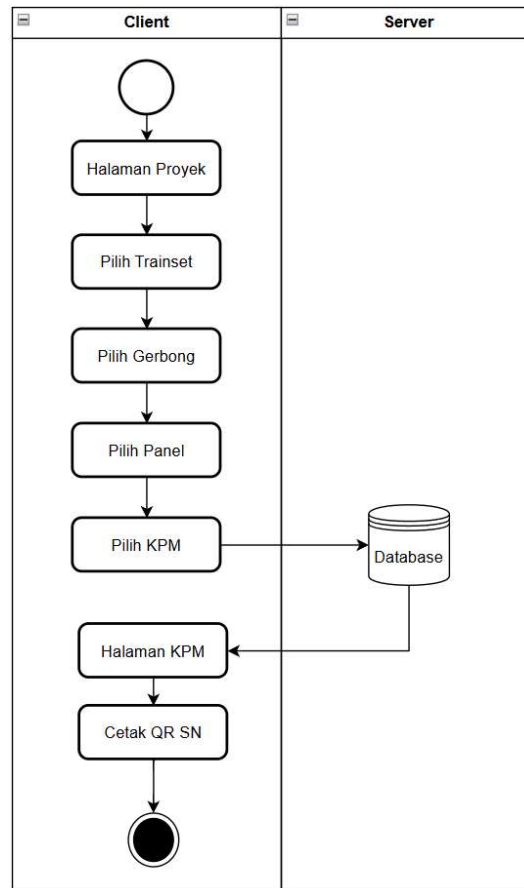


PSPEC 6.2 Cetak QR SN

User melakukan pencetakan QR Serial Number dari sistem

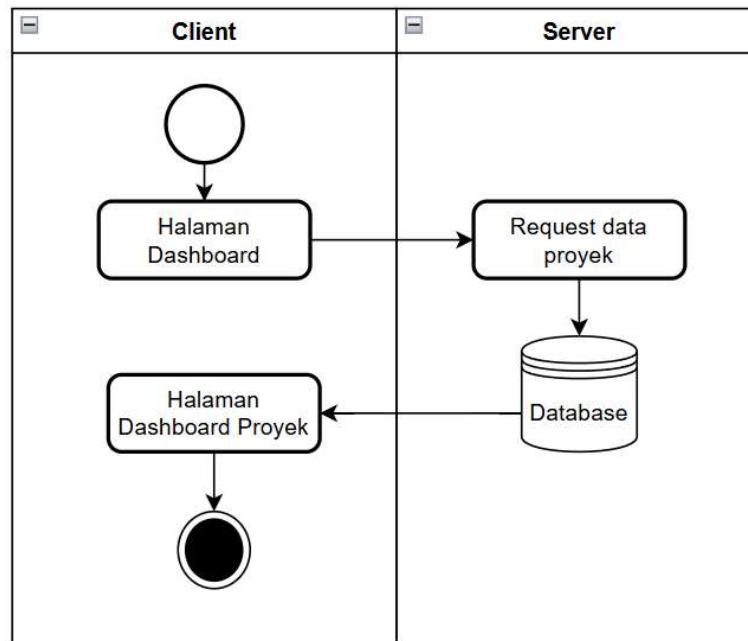
Masukan : proyek, trainset, gerbong, panel

Keluaran : data SN Panel



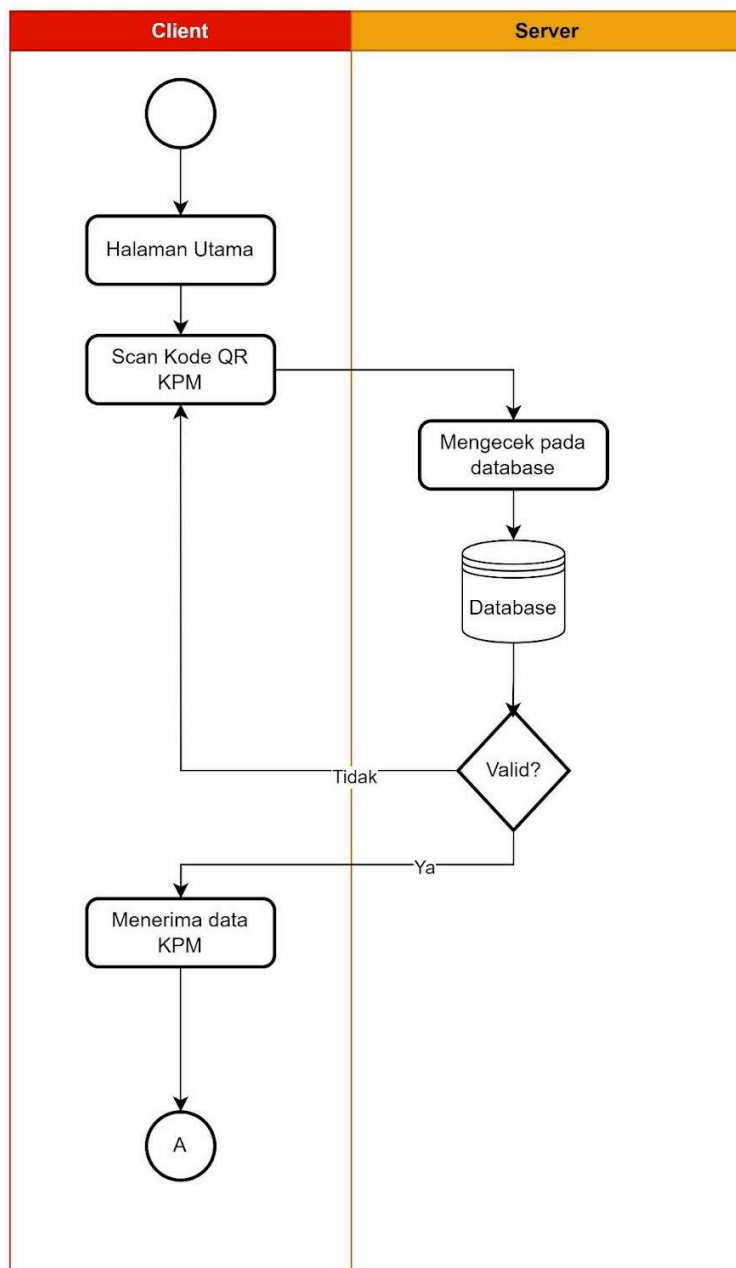
7. Proses Dashboard Proyek

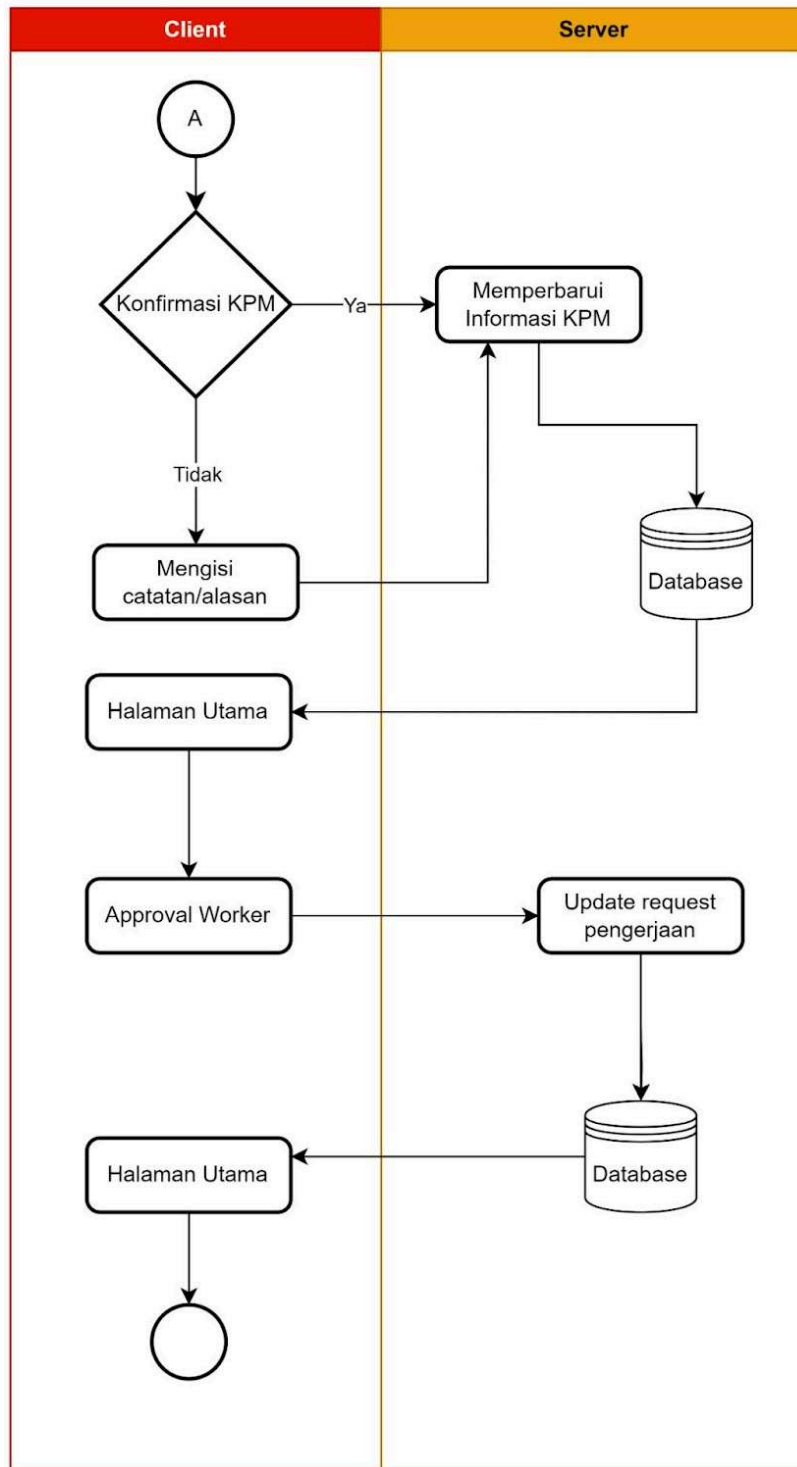
PSPEC 7.1 Dashboard Proyek User mengakses halaman dashboard yang berisi grafik progress produksi proyek
Masukan : -
Keluaran : data proyek



8. Proses Supervisor Mekanik dan Elektrik

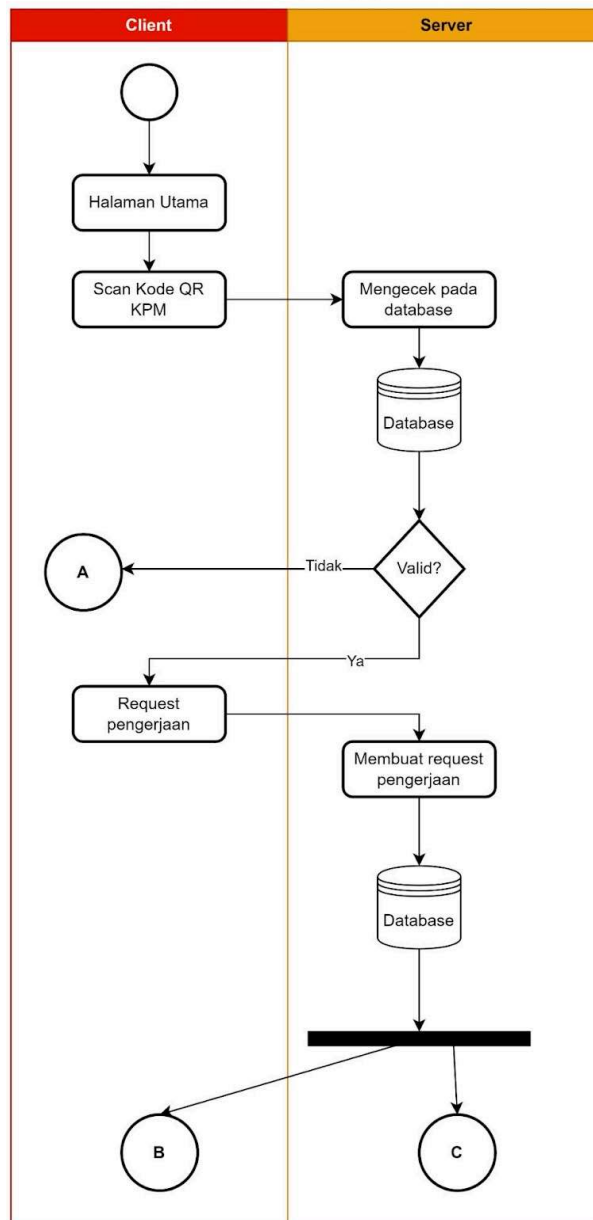
PSPEC 8.1 Supervisor Mekanik / Elektrik User mengakses halaman KPM untuk mengkonfirmasi material yang diterima dan cek request pengerjaan teknisi
Masukan : data jumlah material, data QR
Keluaran : data KPM, data request teknisi

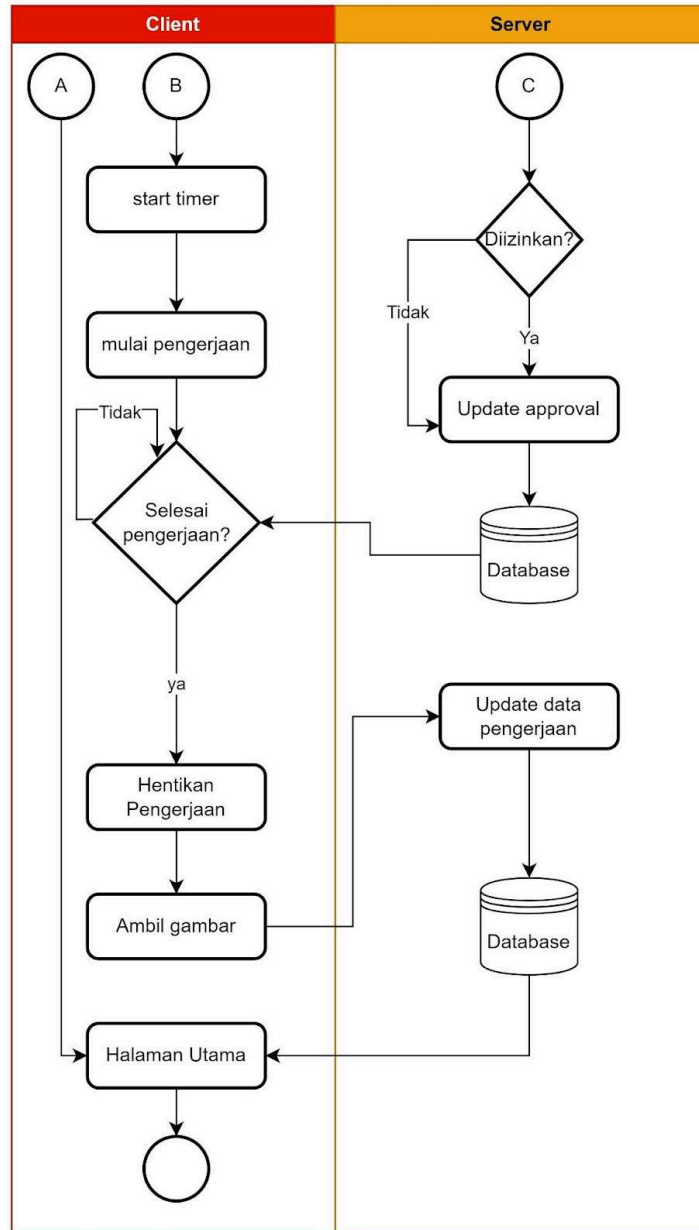




9. Proses Mekanik / Elektrik

PSPEC 9.1 Alur mulai pengerjaan mekanik / elektrik
User scan QR KPM untuk melakukan request mulai pengerjaan.
Ketika menyelesaikan pekerjaan, user mengunggah bukti foto.
Masukan : data QR KPM, identitas user, dokumen foto.
Keluaran : data proses, data KPM





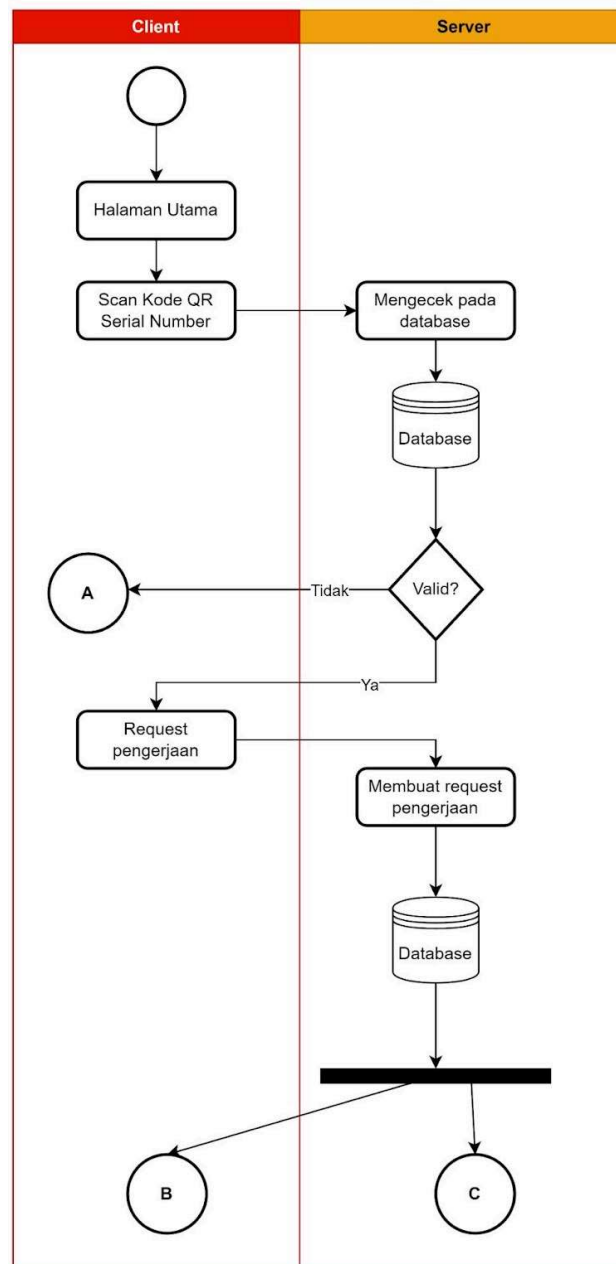
10. Proses Assembly

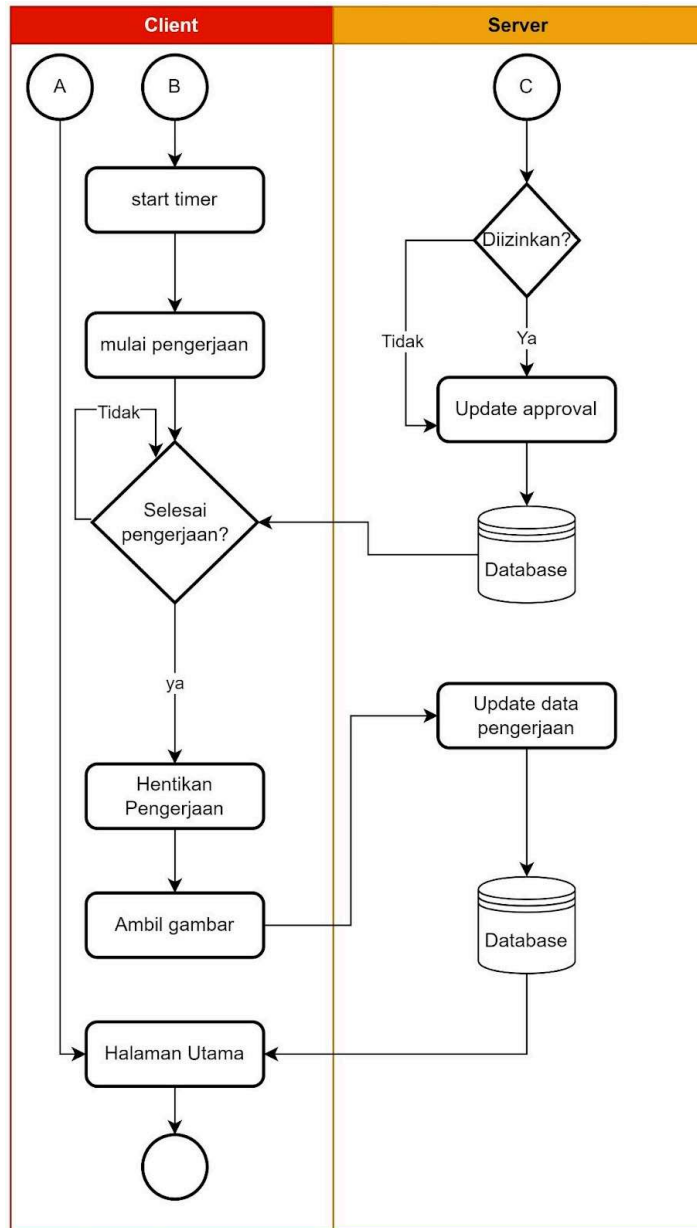
PSPEC 10.1 Assembly

User memindai kode QR SN dan memulai pengerjaan. Lalu mengunggah foto bukti penyelesaian untuk update proses.

Masukan : data QR SN, data foto bukti

Keluaran : data proses, data SN





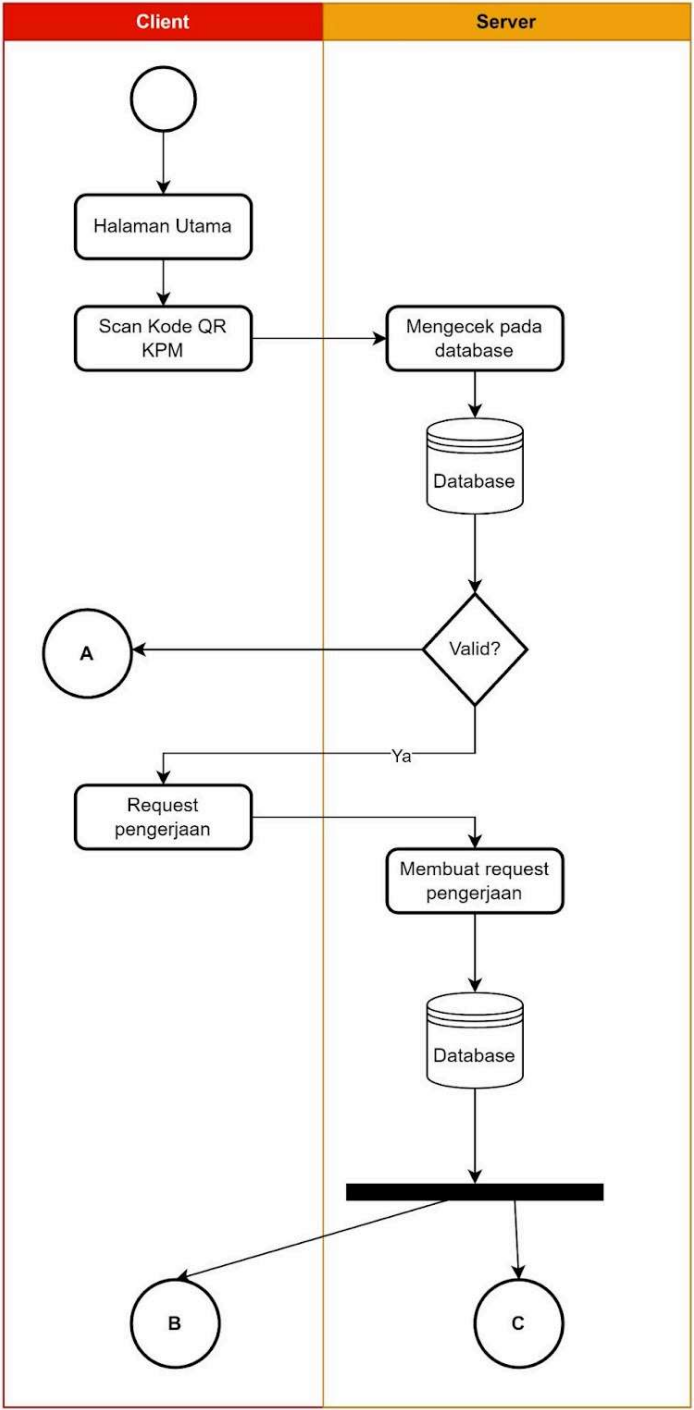
11. Proses QC Proses

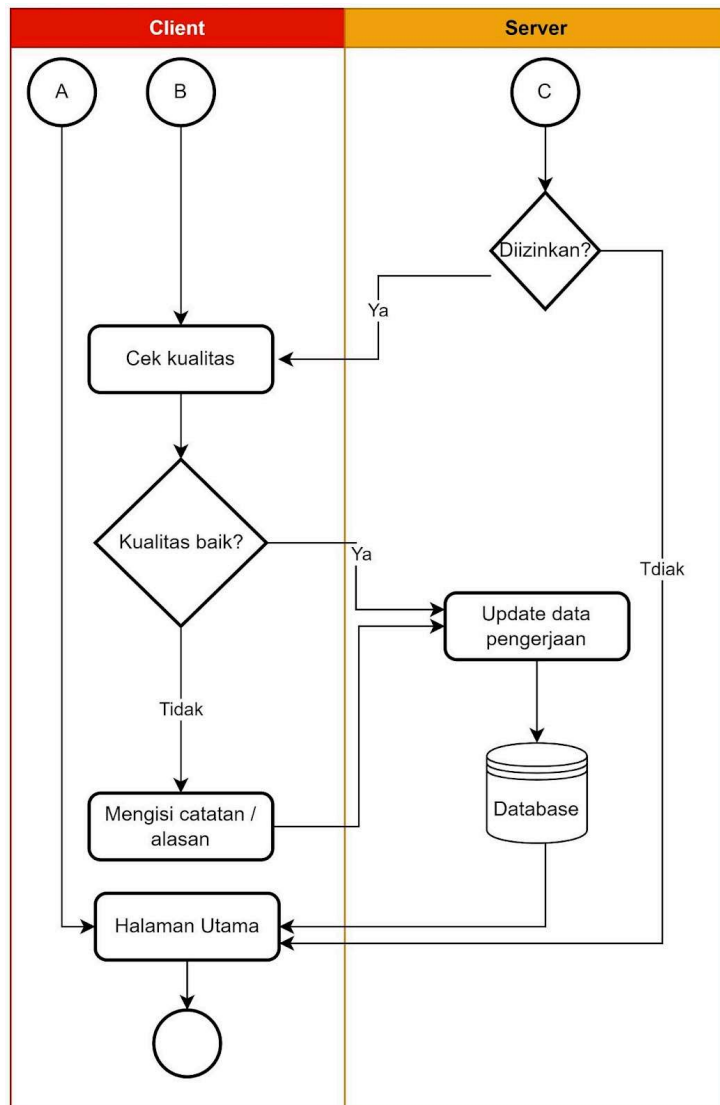
PSPEC 11.1 QC Proses

User memindai kode QR KPM dan memulai pengecekan. Lalu mengupdate proses pengecekan.

Masukan : data QR KPM

Keluaran : data proses, data KPM





12. Proses QC Final

PSPEC 12.1 QC Final

User memindai kode QR SN dan memulai pengecekan. Lalu mengupdate proses penyelesaian.

Masukan : data QR SN

Keluaran : data proses, data SN

