

Nama kontributor	Proses dalam DFD	Level DFD	Aliran Data Masuk	Aliran Data Keluar	Contoh Implementasi
Mahardika Bayu Setya Budi(2211102066)	Mengisi form login	1	Data user	status sukses/gagal	<pre>// Fungsi untuk verifikasi login dan menghasilkan token function login(email, password) { // Mencari pengguna berdasarkan email const user = users.find(user => user.email === email); // Jika pengguna tidak ditemukan if (!user) { return "Email tidak terdaftar"; } // Memeriksa kecocokan password if (user.password !== password) { return "Password salah"; } // Jika berhasil login, buat token akses const token = generateToken(user.email); user.token = token; // Simpan token untuk sesi pengguna return { message: "Login berhasil", token: token // Kirim token akses ke pengguna }; }</pre>

					<pre>// Fungsi untuk menghasilkan token (simulasi) function generateToken(email) { return "token-" + btoa(email + "-" + Date.now()); // Token sederhana, biasanya menggunakan JWT }</pre>
Muham mad Rakan Atha (221110 2062)	register akun	1	Data user	status sukses/gagal	<pre>function buat_akun(nama, email, password) { if (!nama !email !password) { return "Data tidak lengkap"; } // Validasi email const regexEmail = /^[^\s@]+@[^\s@]+\.[^\s@]+\$/; if (!regexEmail.test(em ail)) { return "Email tidak valid"; } // Validasi password minimal 8 karakter if (password.length < 8) { return "Password harus minimal 8 karakter"; } const akunBaru = { id: Date.now(), nama: nama, email: email, password: password, }; console.log("Akun berhasil dibuat:",</pre>

					<pre> akunBaru); return "Akun berhasil dibuat!"; } </pre>
Fauzaan dhiyaa S (221110 2063)	Order PS	1	Data Order	Konfirmasi Order	<pre> function orderPS(idPS, tanggalSewa, durasi, pengguna) { // Memeriksa ketersediaan (contoh simulasi) const psTersedia = cariPS(tanggalSewa, durasi, "Jakarta").find(ps => ps.id === idPS); if (!psTersedia) { return "PS tidak tersedia untuk tanggal dan durasi yang dipilih"; } // Proses pemesanan const pemesanan = { id: Date.now(), idPS: idPS, pengguna: pengguna, tanggalSewa: tanggalSewa, durasi: durasi }; console.log("Pemes anan berhasil:", pemesanan); return "Pemesanan berhasil, konfirmasi telah dikirim!"; } </pre>

Fajar Rizky Gandhi P. (2211102068)	Pembayaran	1	Data Order	Status pembayaran sukses/gagal	<pre> function prosesPembayaran(i dPemesanan, metodePembayaran) { // Validasi metode pembayaran const metodePembayaran Valid = ["transfer_bank", "kartu_kredit"]; if (!metodePembayara nValid.includes(meto dePembayaran)) { return "Metode pembayaran tidak valid"; } // Simulasi penyimpanan pembayaran const pembayaran = { idPemesanan: idPemesanan, metode: metodePembayaran, status: "Sukses" }; console.log("Pemba yaran berhasil diproses:", pembayaran); return "Pembayaran berhasil!"; } </pre>
Khoirul Fazri Nurrokhman (2211102079)	Pengiriman atau Pickup PS	1	Data Order	<ul style="list-style-type: none"> • Jika pengiriman, status pengiriman (dalam proses, dikirim). • Jika pick-up, instruksi 	<pre> function prosesPengirimanAt auPickUp(idPemesa nan, opsi, lokasiPengguna = null, lokasiPickUp = null) { if (opsi === "pengiriman") { </pre>

				terkait lokasi dan waktu pengambilan.	<pre> if (!lokasiPengguna) { return "Lokasi pengguna tidak tersedia untuk pengiriman."; } return kirimPS(idPemesanan, lokasiPengguna); } else if (opsi === "pick-up") { if (!lokasiPickUp) { return "Lokasi pick-up tidak tersedia."; } return pickUpPS(idPemesanan, lokasiPickUp); } else { return "Opsi tidak valid. Pilih 'pengiriman' atau 'pick-up'."; } } // Contoh penggunaan: console.log(prosesPengirimanAtauPickUp(12345, "pengiriman", "Jl. Merdeka, No.10")); console.log(prosesPengirimanAtauPickUp(12345, "pick-up", null, "Store Jakarta Selatan")); </pre>
--	--	--	--	---------------------------------------	---